AD-A060 905 ROYAL AIRCRAFT ESTABLISHMENT FARNBOROUGH (ENGLAND) F/G 9/2

THE SYNCHRONOUS MISSION ANALYSIS PROGRAM SYNMAP.(U)

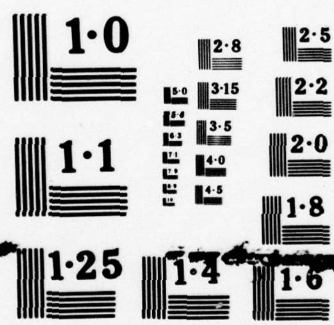
DEC 77 M D PALMER, G E COOK

UNCLASSIFIED RAE-TR-77179 DRIC-BR-64247 NL

1 OF 2

ADA

060905


1·0
2·8
2·5
3·15
2·2
3·5
2·0
4·0
1·1
4·5
1·8
1·25
1·4
1·6

ROYAL AIRCRAFT ESTABLISHMENT

*

Technical Report 77179

December 1977

THE SYNCHRONOUS MISSION ANALYSIS PROGRAM SYNMAP

by

M.D. Palmer

G.E. Cook

*

Procurement Executive, Ministry of Defence

LEVEL II

UDC 629.19.085 : 629.19.014.9 : 629.195 : 521.26 : 5.001.57

14 RAE-TR-77179 18 DRIC

19 BR-64247

ROYAL AIRCRAFT ESTABLISHMENT

9 Technical Report 77179

Received for printing 1 December 1977 11

6 THE SYNCHRONOUS MISSION ANALYSIS PROGRAM SYNMAP

by

10 M. D. Palmer
G. E. Cook

12 108 p.

SUMMARY

A detailed description is given of the computer program SYNMAP, which uses a stochastic simulation method to determine the total velocity increment required for the station acquisition phase of a synchronous satellite orbit mission. The program takes account of errors due to launch vehicle injection, satellite tracking and apogee motor burn. A description is also given of the program POINT2, which may be used to generate the set of random transfer orbits required by SYNMAP.

Departmental Reference: Space 540

310 450

LIST OF CONTENTS

		Page
1	INTRODUCTION	5
2	DESCRIPTION OF POINT2	6
3	INTEGRATION PROCEDURE	8
4	PERTURBATIONS, TIME AND COORDINATE SYSTEMS	9
5	DESCRIPTION OF SYNMAP	9
6	TRACKING ERRORS	12
7	APOGEE MOTOR FIRING STRATEGY	12
	7.1 Fixed drift rate	13
	7.2 Fixed flight path angle	14
	7.3 Optimised flight path angle	14
8	STATION ACQUISITION STRATEGY	15
	8.1 SKYNET 3 station acquisition	16
	8.2 MAROTS station acquisition	16
9	STATISTICS	16
10	INPUT DATA FOR POINT2	17
	10.1 Overall description	17
	10.2 Time card	18
	10.3 First control card	18
	10.4 Second control card	18
	10.5 Nominal orbit card	19
	10.6 Gross attitude manoeuvre card	19
	10.7 Covariance or lower triangular matrix	19
11	INPUT DATA FOR SYNMAP	19
	11.1 Data deck	19
	11.1.1 Control card	20
	11.1.2 Apogee manoeuvre card	20
	11.1.3 Station acquisition card	21
	11.1.4 Perturbation card	21
	11.1.5 Tracking covariance matrix	21
	11.2 Random orbits	21
	11.3 Data stored in program	22
12	SUN-MOON DATA FOR BOTH PROGRAMS	22
13	DESCRIPTION OF SYNMAP OUTPUT	22
Acknowledgment		23
Appendix A	POINT2 program units	25
Appendix B	POINT2 subprogram specifications	26

LIST OF CONTENTS (concluded)

		Page
Appendix C	SYNMAP program units	65
Appendix D	SYNMAP subprogram specifications	67
References		101
Illustrations		Figures 1-8
Report documentation page		inside back cover

1 INTRODUCTION

The program SYNMAP (an acronym for synchronous mission analysis program) is part of the computer software required for analysing the launch phase of a synchronous satellite mission. The program uses a stochastic simulation method to determine the total velocity increment required for station acquisition, taking account of launch vehicle injection errors and apogee motor burn errors. Tracking errors may also be included, but their effect is small and they are normally neglected. The program may be used for any synchronous mission in which the spacecraft is injected into a transfer orbit and then inserted into a drift orbit by use of a fixed impulse apogee motor. Missions for which the program has been used include SKYNET 3 [1] and MAROTS[2].

The program must be supplied with a set of random transfer orbits, each of which is specified by the cartesian components of the geocentric position and velocity vectors at a time shortly before the nominal apogee boost motor (ABM) firing point. These orbits are usually generated by the program POINT2, a modified version of POINT which is described in detail in Ref 3.

SYNMAP provides a choice of strategy for both ABM firing and station acquisition. The firing strategy may optimise the drift orbit flight path angle to minimise the total velocity increment required for station acquisition, provide a given flight path angle or provide a given drift rate. The station acquisition strategy enables the satellite to be placed on station with either an absolute minimum velocity increment or a minimum consistent with the satellite being moved through the smaller longitude range.

The output from SYNMAP consists of two lines for each simulation and a block of statistical information. For each simulation, the program gives the right ascension and declination of the ABM thrust direction, the drift orbit osculating elements immediately after ABM burn, the longitude of the burn point and the delta-velocity components required for station acquisition and circularization of the orbit. The statistical information consists of the mean and standard deviation for the following quantities: drift orbit semi major axis, inclination, eccentricity and right ascension of the ascending mode; ABM firing longitude; right ascension and declination of the ABM thrust direction; solar aspect angle and the total velocity increment required for station acquisition. A table giving a histogram of the velocity increments required is also printed.

The software is written in 1900 FORTRAN for use on ICL 1900 series computers. POINT2 requires approximately 15K words of core store, and consists of

a main program and 18 subprograms, while SYNMAP requires approximately 20K words of core store, and consists of a main program and 32 subprograms. The program units and subprogram specifications for POINT2 are given in Appendices A and B, and those for SYNMAP in Appendices C and D. Their calling structures are illustrated in Figs 1 and 2.

2 DESCRIPTION OF POINT2

The program POINT2 is used to generate two sets of random transfer orbits for use with SYNMAP. One set is for the prime ABM firing apogee and the second for the back-up apogee. Each set has a common epoch, usually three hours before apogee, and each orbit is defined by its geocentric position and velocity vectors at epoch.

The program is provided with a set of nominal launch vehicle injection conditions (velocity, climb angle, azimuth, radial distance, latitude and longitude). These, represented by the elements of the column vector $\underline{x}$, are assumed to have a multivariate normal distribution about their mean vector $\underline{\mu}$, such that the probability density function of $\underline{x}$ is given by

$$f(\underline{x}) = \left[(2\pi)^n|V|\right]^{-\frac{1}{2}} \exp\left[-\tfrac{1}{2}(\underline{x}-\underline{\mu})^T V^{-1}(\underline{x}-\underline{\mu})\right] \qquad (1)$$

where V is the covariance error matrix and n is the number of variates (six in the present case).

Simulation of the injection process requires the generation of random vectors from the distribution defined by equation (1); the stochastically dependent elements of this vector can be constructed from a set of independent normal variables using the linear transformation method suggested by Moonan[4]. Consider a column vector $\underline{z}$ of m independent variables (z_i, $i = 1,\ldots,m$) with

$$E(z_i) = 0 , \qquad 1 \leqslant i \leqslant m$$

$$E(\underline{z}\underline{z}^T) = \underline{I} ,$$

where E is expectation and $\underline{I}$ is the unit matrix of order $m \times m$. $\underline{z}$ may be transformed using the linear mapping

$$\underline{x} - \underline{\mu} = C\underline{z} , \qquad (2)$$

where C is a lower-triangular nonsingular matrix of order $m \times m$. For the new vector $\underline{x}$ to have the probability density function of equation (1),

$$E(x_i) = \mu_i , \qquad 1 \leqslant i \leqslant m$$

$$E\left[(\underline{x} - \underline{\mu})(\underline{x} - \underline{\mu})^T\right] = E\left[C\underline{z}\underline{z}^T C^T\right] = CE(\underline{z}\underline{z}^T)C^T = CC^T = V . \qquad (3)$$

Hence to determine the transformation, the symmetric and positive definite matrix V is decomposed into the product of a lower triangular matrix C and its transpose C^T. The elements (c_{ij}) of C can be determined recursively from the equation,

$$\left.\begin{aligned} c_{i1} &= v_{i1}/v_{11}^{\frac{1}{2}} , && 1 \leqslant i \leqslant m \\ c_{ii} &= \left[v_{ii} - \sum_{k=1}^{i-1} c_{ik}^2\right]^{\frac{1}{2}} , && 2 \leqslant i \leqslant m \\ c_{ij} &= \left[v_{ij} - \sum_{k=1}^{i-1} c_{ik}c_{jk}\right]\Big/c_{jj} , && 1 < j < i \leqslant m \\ \text{and} \quad c_{ij} &= 0 , && j > i . \end{aligned}\right\} \qquad (4)$$

To obtain each random vector $\underline{x}$, six independent normal variates, with zero mean and unit variance, are generated and the transformation of equation (2) applied. If r_1 and r_2 are independent random variables from a uniform distribution defined on the interval (0,1), then

$$\left.\begin{aligned} x_1 &= (-2 \ln r_1)^{\frac{1}{2}} \cos 2\pi r_2 \\ \text{and} \quad x_2 &= (-2 \ln r_1)^{\frac{1}{2}} \sin 2\pi r_2 \end{aligned}\right\} \qquad (5)$$

are a pair of independent random variables from a normal distribution with zero mean and unit variance[5]. The program uses equation (5) with pseudo-random numbers for r_1 and r_2 produced by the mixed congruential method, *ie* using generators of the form

$$n_{i+1} = an_i + c \pmod{m} , \qquad (6)$$

where n_i, a, c and m are all non-negative integers.

A set of geocentric position and velocity components is formed from each vector $\underline{x}$ and the orbit integrated to the first epoch using the method described in section 3. An interpolation is performed and the position and velocity components are written to a disc file. The integration is then continued to the second epoch and the interpolated position and velocity components written to a second disc file.

If required, the integration may be halted at a given apogee and a gross attitude manoeuvre simulated by adding the appropriate velocity increments to the velocity vector. The integration is then restarted and continued as before.

To enable sets of orbits to be reproduced, the random number generator is initialised using constants stored within the program.

3 INTEGRATION PROCEDURE

The equation of motion of an earth satellite may be expressed in vector form as

$$\ddot{\underline{r}} = -\mu r^{-3}\underline{r} + \underline{F} \tag{7}$$

where $\underline{r}$ is the position vector relative to the earth's centre of mass, μ (= GM) is the earth's gravitational constant and $\underline{F}$ is the perturbing acceleration. Orbit development is obtained by the numerical integration of the equations of motion as formulated in the Cowell method. Because of the large variations in the velocity and perturbing accelerations around a highly eccentric orbit, a constant time interval cannot be used if the computation is to be efficient. To overcome this difficulty, the equations are transformed in such a way that a constant step length may be used, *ie* analytical step regulation is introduced. Time t is replaced by the independent variable s, defined by

$$dt/ds = r^k/\kappa \tag{8}$$

where κ is a constant. Merson has discussed[6] the selection of k and κ to give the best results and suggested the use of $k = 3/2$ and $\kappa = \mu^{\frac{1}{2}}$; these values are used here. On changing to the independent variable s, where

$$t' = dt/ds = \mu^{-\frac{1}{2}} r^{\frac{3}{2}} , \tag{9}$$

equation (7) becomes

$$\underline{r}'' = -\underline{r} + \frac{3}{2}\left(\frac{\underline{r} \cdot \underline{r}'}{r^2}\right)\underline{r}' + \frac{r^3}{\mu}\underline{F} . \tag{10}$$

The equation for t' can be differentiated to give

$$t'' = \tfrac{3}{2}(\underline{r} \,.\, \underline{r}')/(\mu r)^{\frac{1}{2}} \quad , \tag{11}$$

so that we have four second-order differential equations (for x'', y'', z'', t''). The numerical integration in both POINT2 and SYNMAP is based on an eighth-order Gauss-Jackson second-sum process. A sixth-order Butcher process is used to set up the difference table required before the second-sum procedure can be started.

4 PERTURBATIONS, TIME AND COORDINATE SYSTEMS

The perturbing accelerations automatically included by both programs are those due to the earth's gravitational potential. The earth's disturbing function includes zonal harmonics up to J_9 and tesseral harmonics up to $J_{4,4}$. The accelerations due to atmospheric drag and the gravitational attractions of the sun and moon may be included if required. For a synchronous transfer orbit, the former is normally included and the latter omitted. Atmospheric density is evaluated using simple analytic formulae with values of exospheric temperature based on the Jacchia 1965 model[7].

Calendar dates are reckoned in Modified Julian days (MJD), which are related to (ordinary) Julian days by the formula.

$$\text{MJD} = \text{JD} - 2400000.5 \quad .$$

The coordinate system used in both programs is that suggested by Kozai[8]. The origin O is at the earth's centre of mass and the Oz axis points to the north pole. Ox lies in the plane of the true equator of date, but instead of pointing to the true equinox of date, it is directed towards a projection of the mean equinox of the epoch 1950.0 (MJD 33281.9234); Oy completes the right-handed system Oxyz .

5 DESCRIPTION OF SYNMAP

An overall flowchart for SYNMAP is shown in Fig 3. The program starts by reading all the input data, carrying out any necessary transformations and setting the program constants which are a function of the data. The two random normal deviate generators are called to set constants from data stored in the program. This ensures that the same random number sequences are produced in consecutive runs, thus enabling results to be meaningfully compared. One generator provides a pair of random normal deviates for simulating ABM burn errors. The other produces six deviates which are used to generate a random

vector from the multivariate normal distribution defined by the tracking error covariance matrix.

The epoch to which the random orbits relate is stored in the disc file holding the orbits. It is read and the stochastic simulation commenced by reading the first orbit to be considered. (A facility exists whereby a simulation may be carried out using every orbit, every i th orbit, or just one specified orbit.)

If tracking errors are to be included, a random vector is generated from the error covariance matrix, using the procedure of section 2. This vector is added to the orbit's geocentric position and velocity components (POSVEL) at epoch.

The orbit is integrated forward until the ABM burn point is reached, *ie* the point at which the transfer orbit and required drift orbit plans intersect. At this point, $\underline{r} \cdot \hat{\underline{h}} = 0$, where $\underline{r}$ is the radius vector and $\hat{\underline{h}}$ a unit vector normal to the required drift orbit plane. The integration steps on either side of this point are identified and an interpolation performed to find the burn time and the transfer orbit POSVEL. A subroutine is called to find the nominal ABM firing direction for the strategy to be employed. The alternatives available are discussed in section 7.

The right-ascension (θ) and declination (ϕ) of the nominal firing direction, denoted by the unit vector $\hat{\underline{s}}$ in Fig 4, are found. The actual firing direction is obtained by adding errors selected from a circular distribution. These errors, ε and β, are shown in Fig 4 and defined as follows. The actual firing direction is assumed to lie on the surface of a cone of half-angle ε whose axis is the nominal firing direction. The angle β is the azimuth angle, measured from true north, and lies in the range 0 to 2π. Since all values of β are equally likely, each one is found by generating a random number in the range 0 to 1 and scaling by 2π.

ε satisfies the Rayleigh distribution, *ie*

$$P(\varepsilon < \varepsilon_0) = 1 - \exp(-\varepsilon_0^2/2\sigma^2)$$

and

$$\varepsilon^2 = a^2 + b^2$$

where a and b are orthogonal with zero mean and variance σ^2. If ε_0 is the value of ε which is only exceeded in n out of m cases, then

$$2\sigma^2 = -\varepsilon_0^2 \ln (n/m) \quad .$$

Individual values of ε are found from the equation

$$\varepsilon = \left[-2\sigma^2 \ln (1 - r)\right]^{\frac{1}{2}}$$

where r is a random variable selected from a uniform distribution defined on the the interval (0,1).

The actual firing direction is assumed to have direction cosine $(x,y,z) = (1,0,0)$ in the axis system whose x-axis lies along the actual firing direction, whose y-axis is in the plane containing the nominal and actual firing directions and whose z-axis completes a right handed set. The direction cosines (x_4,y_4,z_4) of the actual firing direction in the inertial axis system, are obtained by performing a series of four rotations given by the equation:

$$x_1 = x \cos (-\varepsilon) + y \sin (-\varepsilon) \quad ,$$

$$y_1 = y \cos (-\varepsilon) - x \sin (-\varepsilon) \quad ,$$

$$z_1 = z \ ,$$

$$x_2 = x_1 \ ,$$

$$y_2 = y_1 \cos (\beta - \pi/2) + z_1 \sin (\beta - \pi/2) \ ,$$

$$z_2 = z_1 \cos (\beta - \pi/2) - y_1 \sin (\beta - \pi/2) \ ,$$

$$x_3 = x_2 \cos (\pi/2 - \phi) - y_2 \sin (\pi/2 - \phi) \ ,$$

$$y_3 = y_2 \ ,$$

$$z_3 = z_2 \cos (\pi/2 - \phi) + x_2 \sin (\pi/2 - \phi) \ ,$$

$$x_4 = x_3 \cos (-\theta) + y_3 \sin (-\theta) \quad ,$$

$$y_4 = y_3 \cos (-\theta) - x_3 \sin (-\theta) \quad ,$$

and

$$z_4 = z_3 \ .$$

The original unperturbed orbit is now integrated forward to the burn time. The actual ABM velocity increment, $\delta v'$, is given by

$$\delta v' = \delta v + z\sigma_{\delta v}$$

where z is a random normal deviate and $\sigma_{\delta v}$ is the standard deviation of the nominal ABM velocity increment. The satellite's drift orbit components after the burn are given by

$$\underline{v}_d = \underline{v}_b + \delta v'\hat{\underline{s}}$$

where $\underline{v}_d$ and $\underline{v}_b$ are the drift and transfer orbit velocity vectors and $\hat{\underline{s}}$ is a unit vector in the actual ABM firing direction.

A subroutine is called to find the delta-velocity required for station acquisition. The strategy to be employed will be mission dependent. Two possible strategies, available in the program, are described in section 8.

Finally, information required for statistical output is stored and the next simulation begun.

6 TRACKING ERRORS

Tracking errors may be included in the simulation provided that an error covariance matrix is available at epoch. This could be obtained from a tracking analysis program using observations containing errors relating to the actual mission.

The covariance matrix is decomposed into the product of a lower triangular matrix, C , and its transpose, C^T . For each simulation the procedure of section 2 is followed. An error vector $C\underline{z}$ is generated and added to the position and velocity components to give the actual orbit at epoch.

7 APOGEE MOTOR FIRING STRATEGY

The program provides a choice of three apogee motor firing strategies. The ABM may be used to give either a specified drift rate or a specified flight path angle immediately after burn. Alternatively a procedure may be used to optimise the flight path angle so that station acquisition can be achieved with a minimum total velocity increment. The required drift orbit is specified in terms of its inclination (i_d) and right-ascension (Ω_d).

7.1 Fixed drift rate

The drift orbit semi major axis (a), corresponding to the required drift rate (n), is given by the equation

$$a = a_0/(1 + n/n_0)^{\frac{2}{3}}$$

where a_0 is the synchronous orbit radius and n_0 is the earth's rotation rate. The drift orbit velocity (v_d) is given by

$$v_d = [\mu(2/r_b - 1/a)]^{\frac{1}{2}}$$

where μ is the earth's gravitational constant and r_b is the satellite's radial distance at ABM burn. Let the direction cosines of $\underline{v}_d$ and $\hat{\underline{h}}$, a unit vector normal to the drift orbit plane, be (ℓ_x, ℓ_y, ℓ_z) and (h_x, h_y, h_z) respectively. Then $(h_x, h_y, h_z) = (\sin i_d \sin \Omega_d, -\sin i_d \cos \Omega_d, \cos i_d)$. If the direction cosines of the transfer orbit velocity vector $\underline{v}_b$ at the ABM burn point are (b_x, b_y, b_z), then

$$h_x\ell_x + h_y\ell_y + h_z\ell_z = 0 ,$$

$$\ell_x^2 + \ell_y^2 + \ell_z^2 = 1 ,$$

and $\cos\theta = b_x\ell_x + b_y\ell_y + b_z\ell_z = (v_b^2 + v_d^2 - \delta v^2)/2v_bv_d$ where θ is the angle between $\underline{v}_b$ and $\underline{v}_d$.

Therefore
$$\ell_z = -[h_x\ell_x + h_y\ell_y]/h_z ,$$

and

$$b_x\ell_x + b_y\ell_y - b_z\frac{[h_x\ell_x + h_y\ell_y]}{h_z} = \cos\theta$$

ie

$$\left[b_x - \frac{b_zh_x}{h_z}\right]\ell_x + \left[b_y - \frac{b_zh_y}{h_z}\right]\ell_y = \cos\theta$$

or

$$A\ell_x + B\ell_y = \cos\theta .$$

Setting $\ell_x = [\cos\theta - B\ell_y]/A$, we have

$$\ell_x^2 + \ell_y^2 + \left[h_x^2\ell_x^2 + 2h_x\ell_xh_y\ell_y + \ell_y^2h_y^2\right]/h_z^2 = 1 .$$

ie

$$\left[\frac{\cos\theta}{A} - \frac{B}{A}\ell_y\right]^2\left[1 + \frac{h_x^2}{h_z^2}\right] + \frac{2h_x h_y}{h_z^2}\left[\frac{\cos\theta}{A} - \frac{B}{A}\ell_y\right]\ell_y + \ell_y^2\left[1 + \frac{h_y^2}{h_z^2}\right] = 1$$

or

$$C\ell_y^2 + D\ell_y + E = 0 \quad .$$

This equation has two possible solutions. These are evaluated and the corresponding values of ℓ_x and ℓ_y computed. The drift orbit eccentricity for each solution is calculated and the unit vector $\hat{\underline{s}}$ in the ABM firing direction is determined for the solution giving the minimum drift orbit eccentricity using the equation,

$$\delta v\, \hat{\underline{s}} = \underline{v}_d - \underline{v}_b \quad ,$$

where δv is the nominal ABM velocity increment.

7.2 Fixed flight path angle

Fig 5 shows the vectors and angles referred to below. The unit vector $\hat{\underline{u}}$ is defined by the equation

$$\hat{\underline{u}} = \hat{\underline{r}}_b \times \hat{\underline{h}}$$

where $\hat{\underline{r}}_b$ is a unit vector along the radius vector at the ABM burn point. The flight path angle, γ , is measured from $\hat{\underline{u}}$ in the plane defined by $\hat{\underline{u}}$ and $\hat{\underline{r}}_b$. If $\hat{\underline{v}}_d$ is a unit vector along the drift orbit velocity vector $\underline{v}_d$, then

$$\hat{\underline{v}}_d = \hat{\underline{u}}\cos\gamma + \hat{\underline{r}}_b\sin\gamma$$

and

$$|\underline{v}_d| = |\underline{v}_b \cdot \hat{\underline{v}}_d| + \left[\delta v^2 - (v_b^2 - |\underline{v}_b \cdot \hat{\underline{v}}_d|^2)\right]^{\frac{1}{2}} \quad ,$$

$\hat{\underline{s}}$ being found as above.

7.3 Optimised flight path angle

For this strategy, the drift orbit flight path angle is optimised so that the total velocity increment required for station acquisition is a minimum. The angle $\Delta\alpha$, between the transfer and drift orbit planes is given by

$$\cos \Delta_{\alpha} = \sin i_d \sin i_t \cos \Delta\Omega + \cos i_d \cos i_t ,$$

where $\Delta\Omega$ is the required nodal rotation and i_t is the transfer orbit inclination. The angle ϕ_t between the transfer orbit velocity and radius vectors is given by

$$\cos \phi_t = \hat{\underline{v}}_b \cdot \hat{\underline{r}}_b .$$

The angle θ between the drift orbit and transfer orbit velocity vectors is given by

$$\cos \theta = \cos \phi_t \cos \phi_d + \sin \phi_t \sin \phi_d \cos \Delta\alpha$$

where ϕ_d is the angle between the drift orbit radius and velocity vectors. Now, $\delta v^2 = v_d^2 + v_b^2 - 2v_d v_b \cos \theta$ and so, given θ , v_d can be determined. The direction cosines of $\underline{v}_d$, (ℓ_x, ℓ_y, ℓ_z), are obtained using the procedure described in section 7.1.

For each alternative solution, the product $\hat{\underline{v}}_d \cdot \hat{\underline{v}}_b$ is formed and the components of $\underline{v}_d$ found for the scalar product which is closest in value to $\cos \theta$. Sufficient information is available to enable the total delta-velocity required for station acquisition to be calculated.

The program sets an initial value of ϕ_d , (80°), and determines the velocity increment required for station acquisition. The angle ϕ_d is then incremented in small steps until the velocity increment passes through a minimum. The step size is then reduced and the procedure repeated until ϕ_d is known to an acceptable accuracy. The appropriate set of direction cosines are found as described above, and the nominal ABM firing direction derived from them.

8 STATION ACQUISITION STRATEGY

The station acquisition strategy is dependent on the mission requirements. For most missions, a special subroutine will have to be written and incorporated in the program. However, two alternative strategies are included and either may form the basis of a new subroutine. These strategies were the ones required for the SKYNET 3 [1] and MAROTS [2] missions.

The main aim of station acquisition is to achieve a circular synchronous orbit with the satellite at a given longitude. This must be done within a specified time interval and with the minimum expenditure of fuel consistent with any other constraints. For SKYNET 3 there were no other constraints and so the total velocity increment required for station acquisition was minimized. For MAROTS

however, it was required that the satellite should acquire station by the shortest path. Possible constraints for other missions are: (a) moving a satellite so that it is always visible from a given ground station; (b) acquiring station in one direction only and (c) varying the drift rate during the station acquisition sequence.

Whichever strategy is required, the first steps are to find the osculating elements of the drift orbit, the satellite's longitude at ABM burn, its longitude after any tracking period has elapsed and its drift rate relative to the earth. In general, the ABM firing strategy will have been chosen to minimise the drift orbit eccentricity, although a correction may still be required. A manoeuvre may also be required to rotate the orbit plane. This is especially likely for missions which require north-south stationkeeping.

All delta-velocity calculations are made with linearised equations.

8.1 SKYNET 3 station acquisition

The velocity increments needed to acquire station using both easterly and westerly drifts are determined and the direction giving the minimum delta-velocity selected. If the initial drift rate is inadequate for the satellite to reach the specified longitude in the prescribed time, a manoeuvre is performed. The velocity increment required to stop the satellite is then computed. It is assumed that these manoeuvres are also used to reduce the orbital eccentricity. Finally, the velocity increments required to remove any residual eccentricity and perform an inclination correction are calculated.

8.2 MAROTS station acquisition

The drift direction which gives the shortest path to station is identified and the satellite's initial drift rate examined. If it is inadequate, or in the wrong direction, a manoeuvre is performed. A further manoeuvre is made to stop the satellite on station. Again it is assumed that these manoeuvres are also used to reduce the orbital eccentricity. Lastly, the velocity increments required to remove any residual eccentricity and perform an inclination correction are determined.

9 STATISTICS

The program computes the means and standard deviations of the right-ascension and declination of the required ABM firing direction; the solar aspect angle; the spin axis elevation angle (defined as the angle between the actual ABM firing direction and the plane normal to the radius vector at the ABM burn point);

the drift orbit osculating elements; the longitude of the burn point and the total velocity increment (ΔV) required for station acquisition. The following expressions are used to evaluate means and standard deviations:-

$$\bar{x} = \sum_{1}^{i=n} x_i/n$$

and

$$\sigma_x = \left[\frac{\sum_{1}^{i=n} x_i^2}{n} - \bar{x}^2 \right]^{\frac{1}{2}}$$

where n is the number of simulations. A count is also kept of the number of violations of the solar aspect angle constraint.

For representative results to be obtained, at least 250 simulations should be performed. For more detailed analysis, it is suggested that at least 1000 cases be considered.

Since the distributions of some quantities, notably ΔV , are skew, no great significance should be attached to the actual values of mean plus three standard deviations, *ie* $\bar{x} + 3\sigma_x$. (For a normal distribution, the probability of a quantity not exceeding this value is 0.997.)

The program also produces a histogram of the ΔV s required.

10 INPUT DATA FOR POINT2

10.1 Overall description

For convenience, the input data is treated here as a set of 17 punched cards. The order of the cards is as follows:

(i) time card

(ii) control cards

(iii) nominal orbit card

(iv) gross attitude manoeuvre card

(v) covariance matrix or equivalent lower triangular matrix.

All records, except those specifying a matrix, are read in free format. In the description which follows, parameters and constants are referred to by their Fortran-variable names. A typical data deck is illustrated in Fig 6.

10.2 Time card

The time card contains four parameters specifying the injection epoch and the times at which the sets of random orbits are required.

MJDOCH — The MJD number of the epoch at which injection into the transfer orbit takes place.

EP — The time of injection, expressed as a fraction of a day relative to MJDOCH.

TINT1 — The time at which the first set of random orbits is required in hours from epoch.

TINT2 — The time at which the second set of random orbits is required, in hours from epoch. TINT2 = 0.0 if only one set of orbits is required.

10.3 First control card

This card contains three parameters (all integer) which control the working of the program.

NC — The number of orbits required in each set.

NOLOT — If NOLOT = 2, the program reads an error covariance matrix corresponding to a set of injection conditions, with units ft, ft/s and degrees. The units are converted to km, km/s and radians and the matrix decomposed into its lower triangular form. If NOLOT = 1, the lower triangular matrix is read directly.

LSP — If LSP = 1, luni-solar perturbations are included in the integration.

10.4 Second control card

This card contains one control parameter, one parameter which is used as both a data item and a control parameter, and two data parameters.

HN — Although the integration step-length (H) is set in the BLOCK DATA segment, it may be modified to H/HN by setting HN $\neq$ 0.

ARMA — The area-to-mass ratio of the satellite (m^2/kg). If set to 0.0, air-drag terms are excluded from the integration.

SMEAN — The value of the solar 10.7cm radiation flux averaged over three days and measured in units of $10^{-22} Wm^{-2} Hz^{-1}$. If ARMA = 0.0, then SMEAN = 0.0.

<u>SOLBAR</u> The value of the solar 10.7 cm radiation flux averaged over three months and measured in units of $10^{-22} Wm^{-2} Hz^{-1}$. If ARMA = 0.0, then SOLBAR = 0.0.

10.5 <u>Nominal orbit card</u>

The nominal orbit at epoch is supplied as a set of injection conditions:

<u>IPN</u>(6) Velocity; climb angle; azimuth; radial distance; latitude and longitude of injection, in units km, km/s and degrees.

10.6 <u>Gross attitude manoeuvre card</u>

This card contains one parameter which is used both as a control parameter and a data item, and three data parameters.

<u>NAM</u> The apogee number at which the gross attitude manoeuvre is performed. If NAM = 0, no manoeuvre is included.

<u>DVT</u> The transverse delta-velocity component resulting from the manoeuvre (km/s).

<u>DVN</u> The normal delta-velocity component resulting from the manoeuvre (km/s).

<u>DVR</u> The radial delta-velocity component resulting from the manoeuvre (km/s).

10.7 <u>Covariance or lower triangular matrix</u>

The matrix is supplied on 12 cards, each containing three elements punched in 3E15.8 format. Cards one and two hold the first row, cards three and four the second row and so on.

11 <u>INPUT DATA FOR SYNMAP</u>

The input data is divided into three categories. The first is a card data deck containing information about the mission. The second is a set of random orbits stored in an unformatted disc file. The third is data set within the program. A typical card deck is illustrated in Fig 7.

11.1 <u>Data deck</u>

The data deck contains a minimum of four cards, plus an additional 12 cards if tracking errors are to be included. The order of the cards is as follows:-

(i) control card

(ii) apogee manoeuvre card

(iii) station acquisition card

(iv) perturbation card

(v) tracking covariance matrix.

All records are punched in free format except where indicated. Parameters and constants are referred to by their Fortran variable names.

11.1.1 Control card

This card contains five parameters which control the working of the program:

NC Number of cases (*ie* simulations).

STR If STR = 0.0, ABM pointing errors are neglected. Otherwise STR is the starting value for the RANDOMNO function and is a positive odd integer.

NORB If NORB $\leqslant 1$, all random orbits are used. If NORB > 1 and NC = 1, only orbit number NORB is used. If NC = i and NORB = j, then orbits j, 2j, 3j, ..., ij are used.

NM If NM $\neq$ 0, a tracking data covariance matrix is read and tracking errors are included.

NSTATS If NSTATS > 0, statistical data is output.

11.1.2 Apogee manoeuvre card

This card contains the following variables. The first is punched in A4 format and the others in free format.

SUB The name of the subroutine to be used to determine the ABM firing direction.

DELTAV The nominal ABM velocity increment (km s^{-1}).

DIN The required drift orbit inclination (degree).

DRA The required drift orbit right ascension (degree).

GAM If SUB = DR, GAM is the required drift orbit drift rate (degree/s). If SUB = FPA, GAM is the required drift orbit flight path angle (degree). If SUB = OFPA, GAM is not used.

11.1.3 Station acquisition card

This card contains the following variables. The first is punched in A4 format and the others in free format.

SUBS The name of the station acquisition subroutine to be used.

TAC The total time available for station acquisition (days).

TR Tracking time required after ABM burn (days).

PHI Required station longitude (degrees east of Greenwich).

XIL, XIH Minimum and maximum permitted values of drift orbit inclination (degrees).

11.1.4 Perturbation card

This card contains the following variables:

C1 Although the integration step-length, H, is set in the BLOCK DATA segment, it may be modified to H = H/C1 by setting C1 ≠ 0.

LSP If LSP > 0, luni-solar perturbations are included in the integration.

ARMA The area-to-mass ratio of the spacecraft (m^2/kg). If ARMA > 0, air drag terms are included in the integration.

SMEAN The value of the solar 10.7 cm radiation flux averaged over three days and measured in units of 10^{-22} W m^{-2} Hz^{-1}. If ARMA = 0.0, then SMEAN = 0.0.

SOLBAR The value of the solar 10.7 cm radiation flux averaged over three months and measured in units of 10^{-22} W m^{-2} Hz^{-1}. If ARMA = 0.0, then SOLBAR = 0.0.

11.1.5 Tracking covariance matrix

If tracking errors are to be included, this matrix is supplied on 12 cards punched in 3E0.0 format. Cards one and two hold row one, cards three and four row two and so on.

11.2 Random orbits

The epoch and random orbits are supplied in an unformatted disc file. The epoch is specified as a Modified Julian day number (MJD) and a fraction of a day relative to it. Each orbit is specified as a set of geocentric position and velocity components (x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$,) in km and km s^{-1}.

11.3 Data stored in the program

Four data items are set in the program itself. The BLOCK DATA segment sets the maximum and minimum permitted values of solar aspect angle (115^o and 65^o respectively). Data statements in the MASTER set values of ε_0 (in degrees) and the standard deviation of the ABM velocity increment. ε_0 is defined as the value of ε (see section 5) which is only exceeded in three cases out of 1000. The value of $2\sigma^2$ is deduced from ε_0 by setting

$$0.997 = 1 - \exp(- \varepsilon_0^2/2\sigma^2) ,$$

ie

$$2\sigma^2 = \frac{- \varepsilon_0^2}{\ln (0.003)} .$$

The standard deviation of the ABM velocity increment is divided by the nominal velocity increment to give a value in the range 0 to 1.

12 SUN-MOON DATA FOR BOTH PROGRAMS

To enable the solar aspect angle to be calculated and luni-solar perturbations to be included in the integration, a table of sun-moon coordinates, at daily intervals with second and fourth differences, must be supplied in a disc file. A fuller description is given in the specification of subroutine SMPOS in Appendix B.

13 DESCRIPTION OF SYNMAP OUTPUT

A sample line printer output is shown in Fig 8. A heading is first printed, giving information about the case being considered. A minimum of two lines is then output for each simulation. If the solar aspect angle constraint has been violated, a warning is printed to this effect. The next line consists of the case number, right ascension and declination of the actual ABM firing direction, the drift orbit osculating elements, the longitude of the burn point and the satellite's drift rate. This is followed by one line for each drift direction considered for station acquisition. The direction is printed together with the selected drift rate, the delta-velocities required to set up the drift rate and stop the satellite on station, the sum of the last two quantities, the delta-velocity needed to circularize the orbit, the delta-velocity required for inclination correction and the total delta-velocity required for station acquisition. Finally, when the simulation is complete, the statistical information and the histogram of delta-velocities is output.

Acknowledgment

The authors wish to thank A.W. Odell for his helpful advice and assistance on numerical techniques.

Appendix A

POINT2 PROGRAM UNITS

ANGLE	reduces an angle to the range 0 to 2π
ANGL1	reduces an angle to the range $-\pi$ to π
ARANOR	generates random normal deviates
BLOCK DATA	sets certain constants
DEQRSPT	integrates second order differential equations
GATMAN	simulates gross attitude manoeuvre
INFIND[a]	finds a named area on a disc file
INTFRC[a]	converts a number into its integer and fractional parts
IPTOCO	converts injection parameters to coordinates
ITIME[c]	current time in seconds from midnight
LOTMAT	performs triangular decomposition of a matrix
MODAT	computes atmospheric density
ORBIT2	controls orbit integration and interpolation
PDAUXP[b]	auxiliary for DEQRSPT, computes perturbing accelerations
SCPROD[a]	forms scalar product
SMPOS	reads sun/moon coordinates from disc and interpolates
SOLVIN	performs interpolation
TRINV	determines polar coordinates from cartesians (two-dimensional)
UTD4[c]	permits disc/core transfers.

(a) The subprograms INFIND, INTFRC and SCPROD are written in PLAN and are used as semi-compiled segments.

(b) PDAUXP appears in the subprogram (DEQRSPT) which calls it as DAUX. This is organised through subprogram arguments with an EXTERNAL statement in subroutine ORBIT2. The object is to permit substitution of a different subroutine without alteration of the calling program.

(c) The subprograms ITIME and UTD4, though not part of standard FORTRAN, are provided automatically by the 1900 series FORTRAN compilers and are not described here.

Appendix B

POINT2 SUBPROGRAM SPECIFICATIONS

FUNCTION ANGLE

Summary	- The function reduces an angle x (in radians) to the range $0 \leqslant x < \pi$.
Language	- 1900 Fortran.
Author	- Diana W. Scott (April 1969).
Function statement	- FUNCTION ANGLE(X).
Input argument	-
X	Angle x in radians.
Output function	-
ANGLE	- Value of $x \pm 2n\pi$, such that $0 \leqslant \text{ANGLE} < 2\pi$.
Use of COMMON	- None.
Source deck	- 6 cards, including 1 comment card (ICL code).
Local storage used	- 1 real variable.
Subordinate subprograms	- None.
Explanation	- The standard function AMOD is used to give the

fractional part of $x/2\pi$. If this is negative, 2π is added to the result.

FUNCTION ANGL1

Summary - The function reduces an angle x to the range $-\pi < x \leq \pi$.

Language - 1900 Fortran.

Author - Diana W. Scott (April 1969).

Function statement - FUNCTION ANGL1(X)

Input argument -

X — Angle x in radians.

Output function -

ANGL1 — The value of $x \pm 2n\pi$ such that $-\pi <$ ANGL1 $\leq \pi$.

Use of COMMON - None.

Source deck - 7 cards, including one common card (ICL code).

Local storage used - 1 real variable.

Subordinate subprograms - None.

Explanation - The standard function AMOD is used to give the fractional part of $x/2\pi$. If this is greater than π , 2π is subtracted. If it is less than, or equal to, $-\pi$, 2π is added.

SUBROUTINE ARANOR

Summary - The subroutine generates pseudo-random numbers normally distributed with mean zero and standard deviation one.

Language - 1900 Fortran

Author - A.W. Odell (November 1973).

Subroutine statement - SUBROUTINE ARANOR (Z, N).

Input argument -

N — Integer specifying mode of operation and/or number of random numbers; see explanation.

Input and output argument -

Z(1) — Array containing random numbers; see explanation.

Use of COMMON - None.

Source deck - 19 cards (ICL code).

Local storage used - 3 real variables and 1 integer variable.

Subordinate subprograms - The subroutine ITIME (standard ICL 1900 library subroutine).

Explanation - The method used is described in Ref 5. Two sequences of rectangularly distributed random numbers are generated using the equations:-

$$x_{i+1} = 47101x_i + 1410065412 \pmod{2^{31} - 1}$$

$$y_{i+1} = 46893y_i + 2115099118 \pmod{2^{31} - 1}.$$

A sequence of normally distributed random numbers is obtained from

$$z_i = (-2 \log (x_i/(2^{31} - 1)))^{\frac{1}{2}} \sin (2\pi y_i/(2^{31} - 1)).$$

To obtain a set of n random numbers, N may be set to n before entry. On exit from the routine, Z will contain the random numbers. The initial x and y will have been obtained using the computer clock and the set of numbers will not therefore be repeatable.

In order to provide a repeatable set of numbers, or if the subroutine is to be run on a computer without a clock, x and y should be initialised by placing suitable values in Z(1) and Z(2) and calling the subroutine with N = 0. A set of n random numbers can now be obtained in Z by re-entering the subroutine with N set to -n. The use of the subroutine with N negative may also be used to continue a set of numbers produced by the last entry to the subroutine.

BLOCK DATA (POINT)

Summary - The Fortran BLOCK DATA segment is used to set initial values for certain quantities stored in common blocks /PETURB/, /CINTEG/, /CONST/ and /CON/.

Language - 1900 Fortran.

Author - M.D. Palmer (January 1975).

Data in /PETURB/ -

Variable name	Value	Explanation
HALFCD	1.1	Product $\frac{1}{2}C_D$.
RATE	1.1	Ratio of angular velocity of the atmosphere to that of the earth.

Data in /CINTEG/ -

Variable name	Value	Explanation
H	$2\pi/96$	Integration step length.
PD (18)	All elements 0.0	Partial derivatives of position.
PDVEL (18)	All elements 0.0	Partial derivatives of velocity.

Data in /CONST/ -

Variable name	Value	Explanation
EMU	$398601.3 km^3 s^{-2}$	Earth's gravitational constant.
EJ2	1082.637E-6	Earth's second zonal harmonic J_2 .
EJ3	-2.5310E-6	Earth's third zonal harmonic J_3 .
EJ4	-1.6190E-6	Earth's fourth zonal harmonic J_4 .
C22	2.4369E-6	Tesseral harmonic coefficient C_{22} .
S22	-1.4005E-6	Tesseral harmonic coefficient S_{22} .
C33	0.7387E-6	Tesseral harmonic coefficient C_{33} .
S33	1.4343E-6	Tesseral harmonic coefficient S_{33} .
C44	-0.1846E-6	Tesseral harmonic coefficient C_{44} .
S44	0.2508E-6	Tesseral harmonic coefficient S_{44} .
C31	2.0192E-6	Tesseral harmonic coefficient C_{31} .
S31	0.2278E-6	Tesseral harmonic coefficient S_{31} .
C42	0.3444E-6	Tesseral harmonic coefficient C_{42} .
S42	0.7021E-6	Tesseral harmonic coefficient S_{42} .
ERAD	6378.163km	Earth's mean equatorial radius.

Variable name	Value	Explanation
EOMEGA	7.292115147E-5rad/s	Earth's angular velocity.
PREC	6.079E-12rad/s	Earth's precession rate.
SUNMU	1.327127E11km^{3}s^{-2}	Sun's gravitational constant.
SELMU	4902.756km^{3}s^{-2}	Moon's gravitational constant.

Data in /CON/ -

Variable name	Value	Explanation
EJ5	-0.246E-6	Earth's fifth zonal harmonic J_5.
EJ6	0.558E-6	Earth's sixth zonal harmonic J_6.
EJ7	-0.326E-6	Earth's seventh zonal harmonic J_7.
EJ8	-0.209E-6	Earth's eighth zonal harmonic J_8.
EJ9	-0.094E-6	Earth's ninth zonal harmonic J_9.
C43	1.0390E-6	Tesseral harmonic coefficient C_{43}.
S43	-0.1192E-6	Tesseral harmonic coefficient S_{43}.
C41	-0.5175E-6	Tesseral harmonic coefficient C_{41}.
S41	-0.4814E-6	Tesseral harmonic coefficient S_{41}.
C32	0.7783E-6	Tesseral harmonic coefficient C_{32}.
S32	-0.7552E-6	Tesseral harmonic coefficient S_{32}.

Source deck - 18 cards (ICL code).

179

SUBROUTINE DEQRSPT

Summary - The subroutine integrates a set of up to 22 simultaneous second-order differential equations of the form $\ddot{y}_i = f_i(t, y_1, \ldots, y_N, \dot{y}_1, \ldots, \dot{y}_N)$, $i = 1, \ldots, N$, using a Gaussian eighth-order second-sum predictor-corrector[9]. The integration is started using a Butcher sixth-order seven stage Runge-Kutta process[10] to set up the difference table required before the second-sum procedure can take over.

Language - 1900 Fortran.

Authors - A.W. Odell and G.J. Davison (April 1973).

Subroutine statement - SUBROUTINE DEQRSPT (NTRY, DAUX).

Input arguments -

NTRY: 1 for an initialization entry to the subroutine, and 2 for all normal entries (see Explanation).

DAUX: The auxiliary subroutine which evaluates $\ddot{y}_i$.

Output arguments - None.

Use of COMMON - Certain quantities in common block /CINTEG/ are used as follows:-

Input arguments in /CINTEG/ -

NEQ: Number of equations (normally 4 or 22).

H: Integration step size, h (positive or negative).

Input and output arguments in /CINTEG/ -

T: Independent variable, t.

Y(22): Dependent variables, y_i.

YP(22): First derivatives, $\dot{y}_i$.

Y2P(22): Second derivatives, $\ddot{y}_i$, as computed by the auxiliary subroutine DAUX.

Output arguments in /CINTEG/ -

IAUX: Set to -1 following the initialisation entry (NTRY=1), to 0 if T has been changed (during a normal entry), and to 1 otherwise.

Source deck - 218 cards (ICL code).

Local storage used - 10 integer variables, 1 logical variable, 63 real variables and 814 real-array elements.

Subordinate subprograms - The auxiliary subroutine, named in the call to DEQRSPT which takes the role of DAUX.

Explanation - DEQRSPT integrates a set of N($\leqslant$22) simultaneous second-order differential equations of the form $\ddot{y}_i = f_i(t, y_1, \ldots y_N, \dot{y}_1, \ldots \dot{y}_N)$, $i = 1, \ldots, N$, using a Gaussian eighth-order second-sum predictor corrector. The integration is started using a Butcher (6,7) R-K process to set up the difference table required before the second-sum procedure takes over.

Prior to any integration, DEQRSPT must first be called with NTRY = 1. This is the initialization entry in which the step size $h'(=\frac{1}{2}h)$ is set for the Butcher integration and DAUX is called to evaluate $\ddot{y}_i$ at t_0. All subsequent entries are made with NTRY = 2, and one integration step (two, whilst still in the Butcher mode) is performed before control is recurned to the calling program.

If $y_{i,k}$ and $\dot{y}_{i,k}$ are the values of y_i and $\dot{y}_i$ at $t = t_k$ the formulae used in the next Butcher step to evaluate $y_{i,k+1}$, $\dot{y}_{i,k+1}$ and $\ddot{y}_{i,k+1}$, the values of y_i, $\dot{y}_i$ and $\ddot{y}_i$ at $t = t_k + h'$, are:

$$y_{i,k+1} = y_{i,k} + \sum_{s=1}^{7} w_s k_{is} ,$$

$$\dot{y}_{i,k+1} = \dot{y}_{i,k} + \sum_{s=1}^{7} w_s \ell_{is}$$

and

$$\ddot{y}_{i,k+1} = f_i(t_k + h', y_{1,k+1}, \ldots y_{N,k+1}, \dot{y}_{1,k+1}, \ldots \dot{y}_{N,k+1}) , \qquad i = 1, \ldots, N$$

$$\text{where} \quad k_{is} = h' f_i\left(t_k + c_s h', y_{i,k} + \sum_{j=1}^{s-1} a_{sj} \ell_{ij}, \ell_{is} \right),$$

$$\ell_{is} = \dot{y}_{i,k} + \sum_{j=1}^{s-1} a_{sj} k_{ij} , \qquad s = 1, \ldots, N,$$

179 and the remaining quantities are given by the following table

$$
\begin{array}{c|ccccccc}
0 & & & & & & & \\
c_2 & a_{21} & & & & & & \\
c_3 & a_{31} & a_{32} & & & & & \\
c_4 & a_{41} & a_{42} & a_{43} & & & & \\
c_5 & a_{51} & a_{52} & a_{53} & a_{54} & & & \\
c_6 & a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & & \\
c_7 & a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & \\
\hline
 & w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7
\end{array}
\;=\;
\begin{array}{c|ccccccc}
0 & & & & & & & \\
\frac{1}{3} & \frac{1}{3} & & & & & & \\
\frac{2}{3} & 0 & \frac{2}{3} & & & & & \\
\frac{1}{3} & \frac{1}{12} & \frac{1}{3} & -\frac{1}{12} & & & & \\
\frac{1}{2} & -\frac{1}{16} & \frac{9}{8} & -\frac{3}{16} & -\frac{3}{8} & & & \\
\frac{1}{2} & 0 & \frac{9}{8} & -\frac{3}{8} & -\frac{3}{4} & \frac{1}{2} & & \\
1 & \frac{9}{44} & -\frac{9}{11} & \frac{63}{44} & \frac{18}{11} & 0 & -\frac{16}{11} & \\
\hline
 & \frac{11}{120} & 0 & \frac{27}{40} & \frac{27}{40} & -\frac{4}{15} & -\frac{4}{15} & \frac{11}{120}
\end{array}
$$

Given $y_{i,0}$ and $\dot{y}_{i,0}$, $\ddot{y}_{i,0}$ is obtained from DAUX, and the first two Butcher steps of size h' then give $y_{i,1}$, $\dot{y}_{i,1}$ and $\ddot{y}_{i,1}$ at time $t_1 = t_0 + h$. The process is repeated until $y_{i,8}$, $\dot{y}_{i,8}$ and $\ddot{y}_{i,8}$ are obtained after a total of 16 steps (eight calls to the subroutine). The Butcher process is now complete.

On the ninth call (with NTRY = 2) to DEQRSPT the following difference table is constructed for each $\ddot{y}_i \equiv f_i$.

$$
\begin{array}{ccccccccccc}
 & & \ddot{y}_{i,0} & & & & & & & & \\
 & & & \nabla^{1}_{i,1} & & & & & & & \\
 & & \ddot{y}_{i,1} & & \nabla^{2}_{i,2} & & & & & & \\
 & & & \nabla^{1}_{i,2} & & \nabla^{3}_{i,3} & & & & & \\
 & & \ddot{y}_{i,2} & & \nabla^{2}_{i,3} & & \nabla^{4}_{i,4} & & & & \\
 & & & \nabla^{1}_{i,3} & & \nabla^{3}_{i,4} & & \nabla^{5}_{i,5} & & & \\
 & & \ddot{y}_{i,3} & & \nabla^{2}_{i,4} & & \nabla^{4}_{i,5} & & \nabla^{6}_{i,6} & & \\
 & & & \nabla^{1}_{i,4} & & \nabla^{3}_{i,5} & & \nabla^{5}_{i,6} & & \nabla^{7}_{i,7} & \\
\nabla^{-2}_{i,3} & & \ddot{y}_{i,4} & & \nabla^{2}_{i,5} & & \nabla^{4}_{i,6} & & \nabla^{6}_{i,7} & & \nabla^{8}_{i,8} \\
 & \nabla^{-1}_{i,4} & & \nabla^{1}_{i,5} & & \nabla^{3}_{i,6} & & \nabla^{5}_{i,7} & & \nabla^{7}_{i,8} & \\
\nabla^{-2}_{i,4} & & \ddot{y}_{i,5} & & \nabla^{2}_{i,6} & & \nabla^{4}_{i,7} & & \nabla^{6}_{i,8} & & \nabla^{8}_{i,9} \\
 & \nabla^{-1}_{i,5} & & \nabla^{1}_{i,6} & & \nabla^{3}_{i,7} & & \nabla^{5}_{i,8} & & \nabla^{7}_{i,9} & \\
\nabla^{-2}_{i,5} & & \ddot{y}_{i,6} & & \nabla^{2}_{i,7} & & \nabla^{4}_{i,8} & & \nabla^{6}_{i,9} & & \\
 & \nabla^{-1}_{i,6} & & \nabla^{1}_{i,7} & & \nabla^{3}_{i,8} & & \nabla^{5}_{i,9} & & & \\
\nabla^{-2}_{i,6} & & \ddot{y}_{i,7} & & \nabla^{2}_{i,8} & & \nabla^{4}_{i,9} & & & & \\
 & \nabla^{-1}_{i,7} & & \nabla^{1}_{i,8} & & \nabla^{3}_{i,9} & & & & & \\
\nabla^{-2}_{i,7} & & \ddot{y}_{i,8} & & \nabla^{2}_{i,9} & & & & & & \\
 & \nabla^{-1}_{i,8} & & \nabla^{1}_{i,9} & & & & & & & \\
\nabla^{-2}_{i,8} & & \ddot{y}_{i,9} & & & & & & & & \\
 & \nabla^{-1}_{i,9} & & & & & & & & & \\
\nabla^{-2}_{i,9} & & & & & & & & & &
\end{array}
$$

First the known $\ddot{y}_i$ are differenced to give the part of the table to the right of the $\ddot{y}$, and then the first and second sums $\nabla^{-1}_{i,4}$ and $\nabla^{-2}_{i,3}$ are formed, using the equations

$$\nabla^{-2}_{i,3} = h^{-2}y_{i,4} - B_0\ddot{y}_{i,4} - B_2\nabla^{2}_{i,5} - B_4\nabla^{4}_{i,6} - B_6\nabla^{6}_{i,7} - B_8\nabla^{8}_{i,8}$$

and

$$\nabla^{-1}_{i,4} = h^{-1}\dot{y}_{i,4} - A_0\ddot{y}_{i,4} - A_1\nabla^1_{i,5} - A_2\nabla^2_{i,5} - A_3\nabla^3_{i,6} - A_4\nabla^4_{i,6}$$
$$- A_5\nabla^5_{i,7} - A_6\nabla^6_{i,7} - A_7\nabla^7_{i,8} - A_8\nabla^8_{i,8} \ .$$

The remaining ∇^{-2} and ∇^{-1} quantities above the dotted line are defined on the basis that the difference between any entry and the entry above must equal the entry on the right. Actually only $\nabla^{-1}_{i,8}$ and $\nabla^{-2}_{i,8}$ are required explicitly and these are given by

$$\nabla^{-1}_{i,8} = \nabla^{-1}_{i,4} + \ddot{y}_{i,5} + \ddot{y}_{i,6} + \ddot{y}_{i,7} + \ddot{y}_{i,8}$$

and

$$\nabla^{-2}_{i,8} = \nabla^{-2}_{i,3} + 5\nabla^{-1}_{i,4} + 4\ddot{y}_{i,5} + 3\ddot{y}_{i,6} + 2\ddot{y}_{i,7} + \ddot{y}_{i,8} \ .$$

A table containing the coefficients used in the above and following equations is appended.

The ninth normal call to the subroutine continues with an integration step, using the Gaussian (predictor) formulae:-

$$y_{i,9} = h^2\Big(\nabla^{-2}_{i,8} + C_0\ddot{y}_{i,8} + C_1\nabla^1_{i,8} + C_2\nabla^2_{i,8} + C_3\nabla^3_{i,8} + C_4\nabla^4_{i,8}$$
$$+ C_5\nabla^5_{i,8} + C_6\nabla^6_{i,8} + C_7\nabla^7_{i,8} + C_8\nabla^8_{i,8}\Big)$$

and

$$\dot{y}_{i,9} = h\Big(\nabla^{-1}_{i,8} + F_0\ddot{y}_{i,8} + F_1\nabla^1_{i,8} + F_2\nabla^2_{i,8} + F_3\nabla^3_{i,8} + F_4\nabla^4_{i,8}$$
$$+ F_5\nabla^5_{i,8} + F_6\nabla^6_{i,8} + F_7\nabla^7_{i,8} + F^8_{i,8}\Big) \ ,$$

which require only the quantities immediately above the dotted line in the table.

$\ddot{y}_{i,9}$ is then obtained using DAUX and the row of differences under the diagonal line found. This row is then used to obtain corrected values of $y_{i,9}$ and $\dot{y}_{i,9}$ using the equations

$$y_{i,9} = h^2\Big(\nabla^{-2}_{i,8} + D_0\ddot{y}_{i,9} + D_1\nabla^1_{i,9} + D_2\nabla^2_{i,9} + D_3\nabla^3_{i,9} + D_4\nabla^4_{i,9}$$
$$+ D_5\nabla^5_{i,9} + D_6\nabla^6_{i,9} + D_7\nabla^7_{i,9} + D_8\nabla^8_{i,9}\Big)$$

and

$$\dot{y}_{i,9} = h\Big(\nabla^{-1}_{i,8} + E_0\ddot{y}_{i,9} + E_1\nabla^1_{i,9} + E_2\nabla^2_{i,9} + E_3\nabla^3_{i,9} + E_4\nabla^4_{i,9}$$
$$+ E_5\nabla^5_{i,9} + E_6\nabla^6_{i,9} + E_7\nabla^7_{i,9} + E_8\nabla^8_{i,9}\Big) .$$

$\ddot{y}_{i,9}$ is then redetermined, and the row of differences out to $\nabla^8_{i,9}$ under the dotted line recalculated. This row is then used to obtain the final corrected values of $y_{i,9}$ and $\dot{y}_{i,9}$ using the above equations.

<u>Coefficients for A_i, B_i, C_i, D_i, E_i and F_i</u>

i	0	1	2	3	4	5	6	7	8
A_i	$-\frac{1}{2}$	$-\frac{1}{12}$	$\frac{1}{24}$	$\frac{11}{720}$	$-\frac{11}{1440}$	$-\frac{191}{60480}$	$\frac{191}{120960}$	$\frac{2497}{3628800}$	$-\frac{2497}{7257600}$
B_i	$\frac{1}{12}$	0	$-\frac{1}{240}$	0	$\frac{31}{60480}$	0	$-\frac{289}{3628800}$	0	$\frac{317}{22809600}$
C_i	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{19}{240}$	$\frac{3}{40}$	$\frac{863}{12096}$	$\frac{275}{4032}$	$\frac{33953}{518400}$	$\frac{8183}{129600}$	$\frac{3250433}{53222400}$
D_i	$\frac{1}{12}$	0	$-\frac{1}{240}$	$-\frac{1}{240}$	$-\frac{221}{60480}$	$-\frac{19}{6048}$	$-\frac{9829}{3628800}$	$-\frac{407}{172800}$	$-\frac{330157}{159667200}$
E_i	$-\frac{1}{2}$	$-\frac{1}{12}$	$-\frac{1}{24}$	$-\frac{19}{720}$	$-\frac{3}{160}$	$-\frac{863}{60480}$	$-\frac{275}{24192}$	$-\frac{33953}{3628800}$	$-\frac{8183}{1036800}$
F_i	$\frac{1}{2}$	$\frac{5}{12}$	$\frac{3}{8}$	$\frac{251}{720}$	$\frac{95}{288}$	$\frac{19087}{60480}$	$\frac{5257}{17280}$	$\frac{1070017}{3628800}$	$\frac{25713}{89600}$

SUBROUTINE GATMAN

Summary - The subroutine integrates a satellite orbit to the apse at which a gross attitude manoeuvre is to be performed. It interpolates to find the geocentric cartesian components of position and velocity at the apse and then determines the satellite's velocity components immediately after the manoeuvre.

Language - 1900 Fortran.

Author - M.D. Palmer (August 1975).

Subroutine statement - SUBROUTINE GATMAN (I, TAP, X, Y, Z, XD, YD, ZD).

Output arguments -

I — The Modified Julian day number of the gross attitude manoeuvre.

TAP — The time of the manoeuvre, in fractions of a day, relative to I.

X, Y, Z — The satellite's geocentric position components, $\underline{r} = (x, y, z)$, at the time of the manoeuvre (km).

XD, YD, ZD — The satellite's geocentric velocity components, $\underline{v}_m = (\dot{x}_m, \dot{y}_m, \dot{z}_m)$, immediately after the manoeuvre, (km/s).

Use of COMMON - Certain quantities in the common blocks /CINTEG/, /CONST/, /CSMOON/ and /INTP/ are used as follows:-

Input arguments in /CINTEG/ -

MJDOCH — Modified Julian day number of epoch.

PP(3) — Satellite's geocentric position components, $\underline{r} = (x, y, z)$, initially at epoch and subsequently at the latest integration step (km).

PV(3) — Satellite's geocentric velocity components, $\underline{v} = (\dot{x}, \dot{y}, \dot{z})$, initially at epoch and subsequently at the latest integration step (km/s).

PA(3) — Satellite's geocentric acceleration components at the latest integration step (km/s^2).

Input arguments in /CONST/ -

EMU Earth's gravitational constant (km^3 s^{-2}).

Input arguments in /CSMOON/ -

MJDT Modified Julian day number of the current time.

TIMET The current time, in fractions of a day, relative to MJDT.

Input arguments in /INTP/ -

EP The time of epoch, in fractions of a day, relative to MJDOCH.

NAM The number of the apse at which the manoeuvre is to be carried out.

DVT, DVN, DVR The transverse, normal and radial components of delta velocity, resulting from the manoeuvre (km/s)

Output arguments in /CINTEG/ -

XVEL, YVEL, ZVEL The initial values of dx/ds, dy/ds, dz/ds.

TVEL The initial value of dt/ds (s^{-1})

Source deck - 84 cards (ICL code).

Local storage used - 15 real array elements; 39 real variables; and 2 integer variables.

Subordinate subprograms - The subroutines DEQRSPT, MODAT, PDAUXP, SMPOS, TRINV and the functions ANGLE, ANGL1, INFIND, INTFRC, SOLVIN, SCPROD and UTD4.

Explanation - The subroutine starts and controls the integration of a satellite orbit until a specified apse is reached. An interpolation is then performed to find the position and velocity components at the apse, and the satellite's velocity immediately following the manoeuvre is determined.

The variables TVEL(dt/ds), XVEL(dx/ds), YVEL(dy/ds) and ZVEL(dy/ds) are set in terms of the independent variable s. The integration is started by calling DEQRSPT with the statement CALL DEQRSPT (IND, PDAUXP). On the first call, IND = 1 and on the second and subsequent calls IND = 2. After each integration

step, the time elapsed from epoch is calculated in days. The length of the radius vector, r_i , and its rate of change, $\dot{r}_i$, are determined. If $\dot{r}_i \cdot \dot{r}_{i-1} < 0$, there is an apse between the last two integration steps. A counter is incremented at each apse and when the count is equal to the variable NAM, an interpolation is performed to find the satellite's position $r = (x, y, z)$ and velocity $v = (\dot{x}, \dot{y}, \dot{z})$ at the apse.

The equations used for interpolation are:-

$$f(t) = q^3\left[(1 + 3p + 6p^2)f(t_1) + hp(1 + 3p)f'(t_1) + \frac{h^2}{2} p^2 f''(t_1)\right]$$

$$+ p^3\left[(1 + 3q + 6q^2)f(t_2) - h_q(1 + 3q)f'(t_2) + \frac{h^2}{2} q^2 f''(t_2)\right]$$

for position components and

$$f'(t) = 30h^{-1}p^2q^2\left[f(t_2) - f(t_1)\right] + q^2(1 + 5p)(1 - 3p)f'(t_1)$$

$$+ p^2(1 + 5q)(1 - 3q)f'(t_2) + \frac{h}{2} pq^2(2 - 5p)f''(t_1)$$

$$- \frac{h}{2} p^2q(2 - 5q)f'(t_2)$$

for velocity components, where $h = t_2 - t_1$, $p = (t - t_1)/h$, $q = 1 - p$ and $f(t_i)$, $f'(t_i)$, $f''(t_i)$ are the position, velocity and acceleration components at time t_i .

The satellite's velocity $(\dot{x}_m, \dot{y}_m, \dot{z}_m)$ immediately after the gross attitude manoeuvre is given by

$$\dot{x}_m = \dot{x}\left[1 + \frac{\delta v_T}{v}\right] + \left[y\dot{z} - z\dot{y}\right] \frac{\delta v_N}{rv} + x \frac{\delta v_R}{r} ,$$

$$\dot{y}_m = \dot{y}\left[1 + \frac{\delta v_T}{v}\right] + \left[z\dot{x} - x\dot{z}\right] \frac{\delta v_N}{rv} + y \frac{\delta v_R}{r} ,$$

and

$$\dot{z}_m = \dot{z}\left[1 + \frac{\delta v_T}{v}\right] + \left[x\dot{y} - y\dot{x}\right] \frac{\delta v_N}{rv} + z \frac{\delta v_R}{r}$$

where $(\delta v_T, \delta v_N, \delta v_R)$ are the transverse, normal and radial components of delta-velocity resulting from the manoeuvre.

FUNCTION INFIND

Summary - The function finds the location of a named area on a specified disc file.

Language - PLAN for use with 1900 Fortran.

Author - A.W. Odell (July 1973).

Function statement - FUNCTION INFIND (NAME).

Input argument -

NAME: Name of an area on the disc file up to 12 characters in length; must be array element or text (Hollerith) constant.

Output function

INFIND: Number specifying an area on the disc file, 0 if the name is not found in the index.

Use of COMMON - The first 128 integer locations of blank common are used as temporary working space.

Source deck - 37 cards, including 3 comment cards.

Local storage used - 37 words for program, 7 words for data.

Subordinate subprograms - The subroutine UTD4.

Explanation - The subroutine assumes that an index has been set up on the disc file using subroutine INITD and that information has been put in the index using subroutine ADDINF. Associated with each name in the index is a number specifying an area on the disc file. If this name is not found, INFIND is set to 0; otherwise it is set to the associated number.

Remark: A Fortran version of this subroutine exists.

FUNCTION INTFRC

Summary	- The function computes the integral and fractional parts of a number.
Language	- PLAN, for use with 1900 Fortran.
Author	- A.W. Odell (February 1971).
Function statement	- FUNCTION INTFRC (X).
Input and output arguments -	
X	Real number, which is truncated to its fractional part.
Output function	-
INTFRC	Integral part of X.
Use of COMMON	- None.
Source deck	- 23 cards including 3 comment cards (ICL code).
Local storage used	- 18 words for program.
Subordinate subprograms	- None.

Explanation - X is split into its mathematical integral part and its fractional part. For example:

X = - 3.4

I = INTFRC (X)

would result in I being set to -4 and X to 0.6.

A Fortran version of this subroutine is also available.

179

SUBROUTINE IPTOCO

Summary	- The subroutine computes the geocentric cartesian components of the position and velocity of an earth satellite in PROP axes, given the date and time and a set of the standard parameters used to define orbit injection.
Language	- USA Standard Fortran (USAS X3.9 - 1966).
Authors	- G.E. Cook and R. Clarke (October 1972).
Subroutine statement	- SUBROUTINE IPTOCO (X, Y, Z, XDOT, YDOT, ZDOT, V, CA, AZ, R, XLAT, XLONG, MJD, TIME).
Input arguments	-
V	Speed, v .
CA	Climb angle, θ .
AZ	Azimuth of velocity vector, Ψ (measured E of N).
R	Radial distance, r .
XLAT	Geocentric latitude, ϕ .
XLONG	Longitude, λ .
MJD	Modified Julian Day number.
TIME	Time (fraction of a day) such that t is given by MJD + TIME .
Output arguments	
X,Y,Z	Geocentric coordinates of satellite, x, y, z .
XDOT, YDOT, ZDOT	Geocentric components of velocity vector, $\dot{x}, \dot{y}, \dot{z}$.
XLONG	Longitude in inertial axes, λ' .
Use of COMMON	- None.
Source deck	- 22 cards (ICL code).
Local storage used	- 8 real variables.
Subordinate subprograms	- None.

Explanation - Units of length and time are arbitrary, except for the variables MJD and TIME; angles are in radians.

179 The components of position and velocity are evaluated relative to the following coordinate system: the origin O is at the earth's centre and Oz points towards the north pole; Ox lies in the plane of the true equator of date, but instead of pointing towards the true equinox of date it points towards a projection of the mean equinox of the epoch 1950.0 (MJD 33281.9234); Oy completes the right-handed system Oxyz.

The injection conditions are specified relative to an earth-fixed system with longitude measured eastwards from the Greenwich meridian. The longitude λ' relative to the inertial system defined above is found by adding $\hat{\theta}$ to λ, where $\hat{\theta}$ is the 'modified sidereal angle' given by

$$\hat{\theta} = 100.075542 + 360.985612288(t - 33282.0) ,$$

i.e. the origin has been adjusted slightly from 1950.0. (The modified sidereal angle differs from sidereal time only because of the choice of a non-standard reference direction.)

The geocentric components of position and velocity are obtained from the following equations:

$$x = r \cos\phi \cos\lambda'$$

$$y = r \cos\phi \sin\lambda'$$

$$z = r \sin\phi$$

$$\dot{x} = v \{\sin\theta \cos\phi \cos\lambda' - \cos\theta (\cos\Psi \sin\phi \cos\lambda + \sin\Psi \sin\lambda)\}$$

$$\dot{y} = v \{\sin\theta \cos\phi \sin\lambda' + \cos\theta (\sin\Psi \cos\lambda - \cos\Psi \sin\phi \sin\lambda)\}$$

$$\dot{z} = v \{\sin\theta \sin\phi + \cos\theta \cos\Psi \cos\phi\} .$$

SUBROUTINE LOTMAT

Summary - Given a symmetric and positive definite matrix, the subroutine performs triangular decomposition into a lower triangular matrix, which when postmultiplied by its transpose gives the original matrix.

Language - USA Standard Fortran (USAS X3.9 - 1966).

Authors - G.E. Cook and R. Clarke (October 1972)

Subroutine statement - SUBROUTINE LOTMAT (V, C, L).

Input arguments

V(L, L) - Matrix to be decomposed.

L - Number of rows and columns in V.

Output arguments

C(L, L) - The decomposed lower triangular matrix.

Use of COMMON - None.

Source deck - 24 cards, including 3 comment cards (ICL code).

Local storage used - 3 integer variables.

Subordinate subprograms - None.

Explanation - If V is any symmetric and positive definite matrix with elements v_{ij}, there exists a unique lower triangular matrix C, such that $V = CC'$, where C' is the transpose of C. The elements c_{ij} of C are determined recursively from the following equations:

$$c_{i1} = v_{i1}/v_{11}^{\frac{1}{2}} , \qquad 1 \leqslant i \leqslant m$$

$$c_{ii} = \left(v_{ii} - \sum_{k=1}^{i-1} c_{ik}^2\right)^{\frac{1}{2}} \qquad 2 \leqslant i \leqslant m$$

$$c_{ij} = \left(v_{ij} - \sum_{k=1}^{j-1} c_{ik}c_{jk}\right)\bigg/ c_{jj} , \qquad 1 < j < i \leqslant m$$

and

$$c_{ij} = 0 , \qquad j > i .$$

SUBROUTINE MODAT

179

Summary - The subroutine evaluates upper-atmosphere density, density scale height and scale height gradient using a simple analytic model.

Language - 1900 Fortran.

Authors - G.E. Cook and K.J. Tomlinson (April 1969).

Subroutine statement - SUBROUTINE MODAT (HEIT, TINF, TGRADO, DEN, SCALHT, SHGRAD).

Input arguments -

HEIT	Height above the earth's surface, y .
TINF	Exospheric temperature, T_∞ .
TGRADO	Atmospheric temperature gradient dT/dy at the reference altitude, y_0 (120km).

Output arguments -

DEN	Atmospheric density, ρ (g/cm^3).
SCALHT	Density scale height (km).
SHGRAD	Density scale height gradient.

Use of COMMON - None.

Source deck - 46 cards, including 2 comment cards (ICL code).

Local storage used - 24 real array elements, 16 real variables, 1 integer variable.

Subordinate subprograms - None.

Explanation - The values of density and density scale height are obtained from a simple analytic model of the Earth's upper atmosphere. If g_0 denotes the local value of the acceleration due to gravity, the geopotential height above the reference altitude y_0 is defined by

$$\zeta = \int_{y_0}^{y} \{g(y)/g(y_0)\} dy \ , \tag{1}$$

so that

$$\zeta = (y - y_0)(R + y_0)/(R + y) \quad , \tag{2}$$

R being the mean radius of the Earth. The temperature of the atmosphere is represented as a function of geopotential height by the expression

$$T(y) = T_\infty \{1 - a \exp(-\tau\zeta)\} \quad , \tag{3}$$

where T_∞ is the exospheric temperature and a and τ are constants defined by

$$a = 1 - \frac{T(y_0)}{T_\infty} \quad , \tag{4}$$

and

$$\tau = \frac{1}{T_\infty - T(y_0)} \left(\frac{dT}{dy}\right)_{y=y_0} \quad . \tag{5}$$

If the atmosphere is assumed to be in diffusive equilibrium above the reference altitude, the number density n_i of the ith constituent of molecular (or atomic) mass m_i is given by

$$\frac{1}{n_i}\frac{dn_i}{dy} = -\frac{m_i g}{kT} - \frac{1}{T}\frac{dT}{dy}(1 + \alpha) \quad , \tag{6}$$

where k is Boltzmann's constant and α is the thermal diffusion factor.

With the temperature profile (3), equation (6) can be integrated to give

$$n_i = n_i(y_0) \left\{\frac{1 - a}{1 - a \exp(-\tau\zeta)}\right\}^{1+\gamma_i+\alpha} \exp(-\gamma_i\tau\zeta) \quad ,$$

where $\gamma_i = \dfrac{m_i g(y_0)}{\tau k T_\infty}$.

179

The density ρ is given by

$$\rho = \sum_i n_i m_i \ .$$

For the constant boundary conditions at the reference altitude of 120km we use the values assumed by Jacchia[7] in the construction of his static diffusion profiles:

$$T(120) = 355 \text{ K}$$

$$n(N_2) = 4.0 \times 10^{11} \text{ cm}^{-3}$$

$$n(O_2) = 7.5 \times 10^{10} \text{ cm}^{-3}$$

$$n(O) = 7.6 \times 10^{10} \text{ cm}^{-3}$$

$$n(He) = 3.4 \times 10^{7} \text{ cm}^{-3} \ .$$

For hydrogen we also follow Jacchia and take the concentration at 500km to vary with T_∞ according to the relation

$$\log_{10} n(H;\ 500) = 73.13 - 39.40 \log_{10} T_\infty + 5.5 (\log_{10} T_\infty)^2 \ .$$

The thermal diffusion factor α is taken as -0.4 for helium and as zero for the other constituents.

SUBROUTINE ORBIT2

Summary - The subroutine integrates a satellite orbit to a given epoch and interpolates to find the geocentric position and velocity components. If required, the integration may be continued to a second epoch and another interpolation performed. The effects of a gross attitude manoeuvre at an intermediate apse may be included.

Language - 1900 Fortran.

Author - M.D. Palmer (August 1975)

Subroutine statement - SUBROUTINE ORBIT2

Use of COMMON - Certain quantities in the common blocks /CINTEG/, /CONST/, /CSMOON/ and /INTP/ are used as follows:-

Input arguments in /CINTEG/ -

PA(3) Satellite's geocentric acceleration components. $\underline{\dot{v}} = (\ddot{x}, \ddot{y}, \ddot{z})$ at the latest integration step (km/s^2).

Input arguments in /CONST/ -

EMU The earth's gravitational constant (km^3 s^{-2}).

Input arguments in /CSMOON/ -

MJDT Modified Julian day number of the current time.

TIMET The current time, in fractions of a day, relative to MJDT.

Input arguments in /INTP/ -

EP The time of epoch, in fractions of a day, relative to MJDOCH.

TINT1 The time at which the first interpolation is required, measured in days from epoch.

TINT2 The time at which the second interpolation is required, measured in days from epoch. If TINT2 = 0.0, the integration is terminated after first interpolation.

NAM The number of the apse, at which the manoeuvre is required. NAM = 0 if no manoeuvre is to be included.

Input and output arguments in /CINTEG/ -

MJDOCH — On input, the Modified Julian day number of epoch. On output the Modified Julian day number of the apse at which the attitude manoeuvre was performed.

PP(3) — On input the satellite's geocentric position components, $\underline{r} = (x, y, z)$, at epoch. On output, the satellite's position components at the apse at which the attitude manoeuvre was performed (km).

T — On input the time, in seconds, of epoch relative to MJDOCH. On output, the time, in seconds, of the manoeuvre apse relative to MJDOCH.

PV(3) — On input, the satellite's geocentric velocity components, $\underline{v} = (\dot{x}, \dot{y}, \dot{z})$ at epoch. On output, the satellite's velocity components immediately after the manoeuvre (km/s).

Output arguments in /CINTEG/ -

XVEL, YVEL, ZVEL — The initial values of dx/ds, dy/ds and dz/ds.

TVEL — The initial value of dt/ds (s^{-1}).

Source deck - 82 cards (ICL code).

Local storage used - 15 real array elements, 25 real variables and 2 integer variables.

Subordinate subprograms - The subroutines DEQRSPT, GATMAN, IPTOCO, MODAT, PDAUXP, SMPOS, TRINV and the functions ANGLE, ANGL1, INFIND, INTFRC, SOLVIN, SCPROD and UTD4.

Explanation - If a gross attitude manoeuvre is to be included, subroutine GATMAN is called to find the time of the manoeuvre and the satellite's geocentric position and velocity components immediately after it has taken place.

The subroutine starts the integration of the orbit either from the manoeuvre time or from the initial epoch by calling subroutine DEQRSPT with the statement CALL DEQRSPT (IND, PDAUXP). On the first call IND = 1 and on the second and subsequent calls IND = 2.

After each integration step, the time elapsed from epoch (T2) is calculated and compared with TINT1. If T2 < TINT1 the time, position, velocity and acceleration components are stored and another integration step performed. (To reduce computation time, this storage is not carried out if T2 < 0.9 TINT1.)

When T2 ≥ TINT1, an interpolation is carried out using the equations:-

$$f(t) = q^3\left[(1 + 3p + 6p^2)f(t_1) + hp(1 + 3p)f'(t_1) + \frac{h^2}{2}p^2f''(t_1)\right]$$

$$+ p^3\left[(1 + 3q + 6q^2)f(t_2) - hq(1 + 3q)f'(t_2) + \frac{h^2}{2}q^2f''(t_2)\right]$$

for position components and

$$f'(t) = 30h^{-1}p^2q^2[f(t_2) - f(t_1)] + q^2(1 + 5p)(1 - 3p)f'(t_1)$$

$$+ p^2(1 + 5q)(1 - 3q)f'(t_2) + \frac{h}{2}pq^2(2 - 5p)f''(t_1)$$

$$- \frac{h}{2}p^2q(2 - 5q)f''(t_2)$$

for velocity components, where $h = t_2 - t_1$, $p = (t - t_1)/h$, $q = 1 - p$ and $f(t_i)$, $f'(t_i)$, $f''(t_i)$ are the position, velocity and acceleration components at time t_i.

$f(t)$ and $f'(t)$ are then written to an unformatted disc file on channel 2. If TINT2 ≠ 0.0, the above process is repeated.

SUBROUTINE PDAUXP

Summary - The subroutine calculates the cartesian components of the geocentric accelerations due to the gravitational attractions of the earth, sun and moon, atmospheric drag and the precession of the earth's polar axis. It will also, if required, evaluate the partial derivatives of the acceleration components with respect to the initial components of the position and velocity at epoch.

Language - 1900 Fortran.

Author - M.D. Palmer (November 1974).

Subroutine statement - Subroutine PDAUXP.

Use of COMMON - Certain quantities in the common blocks /CINTEG/, /CON/, /CONST/, /CSMOON/ and /PETURB/ are used as follows:

Input arguments in /CINTEG/ -

MJDOCH	Modified Julian day number of epoch.
NEQ	Number of equations being integrated (22 if partial derivatives of acceleration are required, 4 otherwise).
IAUX	Flag, see explanation.
X, Y, Z	Cartesian position components $(x,y,z) = \underline{r}$.
PD(18)	Partial derivatives of position with respect to the initial position $(\underline{r_0})$ and velocity (v_0); $\partial\underline{r}/\partial(\underline{r_0},\underline{v_0})$.
XVEL, YVEL, ZVEL	The latest values of dx/ds, dy/ds and dz/ds .
T	Time, t , relative to MJDOCH (seconds).
PDVEL(18)	Derivatives with respect to the independent variable s of the partial derivatives of position $$\frac{d}{ds}\left[\frac{\partial\underline{r}}{\partial(\underline{r_0},\underline{v_0})}\right] .$$

Input arguments in /CON/ -

EJ5, EJ6, EJ7, EJ8, EJ9 — Coefficients of the earth's zonal harmonics, $J_5, J_6, \ldots, J_9$.

C32, S32; C41, S41; C43, S43 — Fully normalized coefficients of certain tesseral harmonics, viz. $\overline{C_{32}}, \overline{S_{32}}, \overline{C_{41}}, \overline{S_{41}}, \overline{C_{43}}, \overline{S_{43}}$.

Input arguments in /CONST/ -

EMU — Earth's gravitational constant, μ_e .

EJ_2, EJ_3, EJ_4 — Coefficients of the earth's zonal harmonics, J_2, J_3, J_4 .

C22, S22; C31, S31; C33, S33; C42, S42; C44, S44 — Fully normalized coefficients of certain tesseral harmonics, viz. $\overline{C_{22}}, \overline{S_{22}}, \overline{C_{31}}, \overline{S_{31}}, \overline{C_{33}}, \overline{S_{33}}, \overline{C_{42}}, \overline{S_{42}}, \overline{C_{44}}, \overline{S_{44}}$.

ERAD — Mean equatorial radius of the earth, R .

EOMEGA — Mean rotation rate of the earth's polar axis, in rad/s.

SUNMU — Sun's gravitational constant, μ_s .

SELMU — Moon's gravitational constant, μ_m .

Input arguments in /CSMOON/ -

XS, YS, ZS — Cartesian components of the sun's position ($\underline{r_s}$) at the current time, computed if IAUX < 0 .

XM, YM, ZM — Cartesian components of the moon's position ($\underline{r_m}$) at the current time, computed if IAUX < 0 .

Input arguments in /PETURB/ -

SMEAN — Mean value over three days of the solar 10.7cm radiation flux in units of 10^{-22}W m^{-2} Hz^{-1}.

SOLBAR — Mean value over three solar rotations of the 10.7cm radiation flux in units of 10^{-22}W m^{-2} Hz^{-1}.

RATE — Angular velocity of the atmosphere, ω_a in rad/s.

ARMA — The product AM × HALFCD × 1000 where AM is the satellite's area-to-mass ratio and HALFCD is the quantity ½CD where CD is the satellite's drag coefficient. If ARMA = 0.0 , drag terms are excluded.

TNITE — Minimum night-time temperature (K).

LSP — If LSP > 0 , luni-solar perturbations are included.

NSR — If NSR > 0 , solar radiation pressure terms are included *but* see explanation.

<u>Output arguments in /CINTEG/</u> -

XACC, YACC, ZACC — Values of d^2x/ds^2, d^2y/ds^2, d^2z/ds^2 at the current time.

TVEL — Value of dt/ds at the current time.

PDACC(18) — Second derivatives with respect to the independent variable s of the partial derivatives of position

$$\frac{d^2}{ds^2}\left[\frac{\partial \underline{r}}{\partial(\underline{r_0},\underline{v_0})}\right] .$$

XVELT, YVELT, ZVELT — Cartesian components of velocity, $\dot{x}$, $\dot{y}$, $\dot{z}$, at the current time.

XACCT, YACCT, ZACCT — Cartesian components of acceleration, $\ddot{x}$, $\ddot{y}$, $\ddot{z}$, at the current time.

PDACCT(18) — Partial derivatives of acceleration with respect to the initial position and velocity at epoch

$$\frac{d^2}{dt^2}\left[\frac{\partial \underline{r}}{\partial(\underline{r_0},\underline{v_0})}\right] .$$

<u>Output arguments in /CSMOON/</u> -

MJDT — Modified Julian day number of the current time.

TIMET — The current time, in fractions of a day, relative to MJDT.

<u>Source deck</u> - 155 cards (ICL code).

<u>Local storage used</u> - 51 real variables and 3 integer variables.

<u>Subordinate subprogram</u> - The subroutines MODAT, SMPOS and TRINV and the functions ANGLE, ANGL1, INFIND and SCPROD.

<u>Explanation</u> - When the subroutine is called for the first time, or for a change of epoch, IAUX must be set to -1. For subsequent entries, when the time has changed since the previous entry, IAUX should be set to zero. If the

time has not changed IAUX may be set to 1 to prevent recomputation of certain terms. (All these cases are allowed for by the calling segment in POINT.)

The cartesian components of the geocentric acceleration acting on the satellite are found by adding contributions from the gravitational attractions of the earth, sun and moon, atmospheric drag and the precession of the earth's polar axis. The subroutine is structured so as to permit the easy insertion of solar radiation pressure at a future date.

The earth harmonic terms are:

$$\mu_e \sum_{n,m} R^n \Big/ r^{n+2} \Re\left[E_{n,m}\left\{W^m\left(P_n^{(m+1)}\underline{\hat{z}} - P_{n+1}^{(m+1)}\underline{\hat{r}}\right) + mW^{m-1}e^{-j\theta}P_n^{(m)}(\underline{\hat{x}} + j\underline{\hat{y}})\right\}\right]$$

where the sidereal angle $\theta = \theta_0 + \omega_e t$,

$W = (x + jy)/re^{j\theta})$,

$P_n^{(m)}$ is the mth derivative of the Legendre polynomial, $P_n(z/r)$

$E_{0,0} = 1$,

$E_{n,0} = -J_n$

and

$$E_{n,m} = \left[2(2n + 1)(n - m)! \big/ (n + m)!\right]^{\frac{1}{2}}(\bar{C}_{n,m} - j\bar{S}_{n,m}) \; .$$

Terms are included up to the ninth zonal harmonic and the 4,4 tesseral harmonic.

The accelerations due to the gravitational attractions of the sun and moon are given by

$$\mu_b\left[(\underline{r_b} - \underline{r}) \big/ |\underline{r_b} - \underline{r}|^3 - \underline{r_b} \big/ |\underline{r_b}|^3\right]$$

b being 's' for the sun and 'm' for the moon.

Atmospheric drag is included if ARMA $\neq 0$. A modified version of the Jacchia 1965 model[7] is used to find the ambient air density. Firstly the subroutine computes the satellite's height and latitude (ϕ), the declination of the sun (δ) and the hour angle (H) of the satellite relative to the sun. The exospheric temperature (T_∞) is determined using the equation,

$$T_\infty = T_{NITE}\left[1 + 0.28\theta + 0.28\left[\cos\left\{\frac{\phi - \delta}{2}\right\}^{2.5} - \theta\right]\left[\cos\frac{\tau}{2}\right]^{2.5}\right] ,$$

where $\theta = \sin\dfrac{[\phi + \delta]^{2.5}}{2}$

and $\tau = \left[H - 0.78539816 + 0.20943951 \sin\left[1 + 0.78539816\right]\right]$.

The atmospheric temperature gradient at the reference altitude (120km) is given by:

$$T_{grad_0} = (T_\infty - 355)(0.029 \exp[- x^2/2])$$

where $x = \dfrac{T_\infty - 800}{750 + 1.722 \times 10^{-4}(T_\infty - 800)^2}$.

Subroutine MODAT is called to determine the atmospheric density ρ . The force acting on the satellite is given by $\frac{1}{2}\rho|\underline{V}|^2 SC_D$ where C_D is the drag coefficient, S is the effective cross-sectional area perpendicular to the air flow and $\underline{V}$ is the velocity of the satellite relative to the ambient air. $\underline{V}$ is given by

$$\underline{V} = \underline{v} - \underline{\omega}_a \times \underline{r}$$

where $\underline{r}$ and $\underline{v}$ are the position and velocity vectors of the satellite.

If the cartesian components of $\underline{V}$ are V_x, V_y, V_z then the contributions to the acceleration components are:

$$\ddot{x} = -\rho\frac{SC_D}{2M}|V|\underline{V_x} , \qquad \text{etc.}$$

where M is the mass of the satellite.

The variable ARMA is the quantity $\dfrac{SC_D}{2M} \times 1000$, the 1000 being a unit conversion factor.

The contribution to the acceleration due to the precession term is

$$p(\dot{x}\hat{z} - \dot{z}\underline{\hat{x}}) .$$

179

If required, the subroutine also evaluates the partial derivatives of the accelerations with respect to the initial position ($\underline{r}_0$) and velocity ($\underline{v}_0$) at epoch, i.e.

$$\frac{d^2}{dt^2}\left[\frac{\partial \underline{r}}{\partial(\underline{r}_0, \underline{v}_0)}\right] .$$

FUNCTION SCPROD

Summary - The function gives the scalar (inner) product of two arrays.

Language - PLAN, for use with 1900 Fortran.

Author - A.W. Odell (March 1973).

Function statement - FUNCTION SCPROD (A, B, NA, NB, N).

Input arguments -

A(1), B(1) — Locations of the first elements to be multiplied (must be array elements).

NA, NB — Increments in the arrays, A, B of elements to be multiplied.

N — Number of elements to be multiplied.

Output function -

SCPROD — The scalar product $\sum_{i=1}^{N} A(1 + (i - 1)NA)B(1 + (i - 1)NB)$.

Use of COMMON - None.

Source deck - 25 cards, including 1 comment card (ICL code).

Local storage used - 2 words for variables, plus 22 words for program.

Subordinate subprograms - None.

Explanation - The arrays in the calling routine may have any dimensions although they are treated as one-dimensional arrays in the function. For simplicity A and B are dimensioned '1'. This may cause the function to fail in TRACE2, so it should not be compiled in this mode. If $N \leq 0$, the function will return zero.

SUBROUTINE SMPOS

Summary - The subroutine computes the geocentric cartesian coordinates (km) of the sun and moon at a given time.

Language - ASA Fortran (Standard Fortran 4).

Author - A.W. Odell (May 1973).

Subroutine statement - SUBROUTINE SMPOS.

Dummy arguments - None.

Use of COMMON - Certain quantities in common block /CSMOON/ are used as follows:

Input arguments in /CSMOON/ -

MJDT	Modified Julian day number of the required time.
TIMET	Time, as a fraction of a day, since 0 hours on MJDT.

Output arguments in /CSMOON/ -

XS, YS, ZS	The cartesian coordinates $(x_s, y_s, z_s) = r_s$ of the sun.
XM, YM, ZM	The cartesian coordinates $(x_m, y_m, z_m) = r_m$ of the moon.

Input and output argument in /CSMOON/ -

TABLE(43)	Working space.

Source deck - 27 cards including 1 comment card (ICL code).

Local storage used - 2 integer variables, 1 logical variable.

Subordinate subprograms - The subroutine UTD4, and the functions INFIND and SCPROD.

Explanation - The subroutine obtains the sun-moon coordinates by interpolation in a table of daily positions, with 2nd and 4th differences, using Everett's interpolation formula. The table is read from a disc file when required, using the subroutine UTD4. It is stored in an area called SUNMOONTABLE, the location of which is found using the function INFIND. The area is contained in a disc file, which must have been opened, before entry to SMPOS, on channel 7.

The table is stored as follows: firstly, the modified Julian day numbers of the first and last sets of coordinates (2 integers), then data for each midnight as follows (sets of 18 reals): r_s, r_m, $\Delta^2 r_s$, $\Delta^2 r_m$, $\Delta^4 r_s$, $\Delta^4 r_m$.

If f_0, f_1 denote values of f at $t = 0$ and $t = 1$, Everett's interpolation formula to the 4th differences[11] gives:

$$f(t) = D_0 + t\Big(D_1 + (t-1)\Big(D_2 + (t+1)\Big(D_3 + (t-2)\Big(D_4 + (t+2)D_5\Big)\Big)\Big)\Big),$$

where $D_{2n} = \delta^{2n} f_0/(2n)!$, $D_{2n+1} = \delta^{2n}(f_1 - f_0)/(2n+1)!$.

The data was originally obtained on punched cards from JPL[12,13] and has been transformed, before storing on the disc, into the SAO/PROP[14] system of axes, using the subroutine AX1950. If the time input to SMPOS lies outside the range of data stored on the disc, a STOP77 statement will be obeyed.

FUNCTION SOLVIN

Summary - Given a function and its derivative at two points, the function solves four problems involving cubic interpolation.

Language - ASA Fortran (Standard Fortran 4).

Author - A.W. Odell (January 1975).

Function statement - FUNCTION SOLVIN (MODE, M, FO, F1, FD0, FD1, F).

Input arguments -

MODE Number specifying problem to be solved, see explanation.

H Difference, h, between the arguments of the function (i.e. $h = x_1 - x_0$).

F0, F1 Function values f_0 at $x = x_0$ and f_1 at $x = x_1$.

FD0, FD1 Derivative values f'_0, at $x = x_0$ and f'_1 at $x = x_1$.

F Required function value or argument - see explanation.

Output function -

SOLVIN Solution to problem - see explanation.

Use of COMMON - None.

Source deck - 33 cards, including 5 comment cards (ICL code).

Local storage used - 9 real variables.

Subordinate subprograms - None.

Explanation - A cubic polynomial $P(x) = f_0 + f'_0x + a_2x^2 + a_3x^3$ is first fitted to the data giving $a_2 = (3B - A)/h$ and $a_3 = (A - 2B)/h^2$ where $A = f' - f_0$ and $B = (f_1 - f_0)/h - f'_0$.

Then,

(1) if MODE = 1, Newton's method is used to find the value of x for which $P(x) = F$,

(2) if MODE = 2, Newton's method is used to find the value of x for which $P'(x) = F$,

(3) if MODE = 3, $P(F)$ is evaluated, and

(4) if MODE = 4, $P'(F)$ is evaluated.

SUBROUTINE TRINV

Summary - The subroutine obtains polar coordinates from (two-dimensional) cartesian coordinates.

Language - ASA Fortran (Standard Fortran 4).

Author - R.H. Gooding (May 1968).

Subroutine statement - SUBROUTINE TRINV (Y, X, R, TH).

Input arguments -

Y Cartesian y-coordinate (arbitrary).

X Cartesian x-coordinate (arbitrary).

Output arguments -

R Polar r-coordinate.

TH Polar θ-coordinate.

Use of COMMON - None.

Source deck - 8 cards, including 2 comment cards (ICL code).

Local storage used - None.

Subordinate subprograms - None.

Explanation - The equations $r \cos \theta = x$, and $r \sin \theta = y$ are solved for r and θ, using the standard ATAN2 function to give θ between $-\pi$ and $+\pi$. If $x = y = 0$ the ATAN2 function is not used and θ is set to zero.

Remarks -

(1) The subroutine, if used twice, provides a convenient solution of the three-dimensional cartesian-to-polar transformation. Thus to solve the equations,

$$r \sin \theta \cos \phi = x ,$$

$$r \sin \theta \sin \phi = y ,$$

and

$$r \cos \theta = z$$

for r, θ and ϕ, the following two statements will suffice:

CALL TRINV (Y, X, RSINTH, PHI) , giving $r \sin \theta$ and ϕ

and

CALL TRINV (RSINTH, Z, R, TH) , giving r and θ.

Moreover, this solution will give maximum accuracy for θ and ϕ, with both angles set in the correct quadrant.

(2) The actual input and output arguments must be distinct.

Appendix C

SYNMAP PROGRAM UNITS

ANGLE[a]	reduces an angle to the range 0 to 2π
ANGL1[a]	reduces an angle to the range $-\pi$ to π
ANST[b]	performs station acquisition by the shortest path
APMAN	determines the ABM burn point and the nominal ABM firing direction
ARANOR2[c]	generates a pair of random normal deviates
ARANOR6[c]	generates six random normal deviates
BLOCK DATA	sets certain constants
COTOEL	converts coordinates to osculating elements
DR[d]	Finds the nominal ABM firing direction for a specified drift orbit drift rate
DEQRSPT[a]	integrates second order differential equations
DVST[b]	performs station acquisition with minimum delta-velocity expenditure
EAFKEP	determines eccentric anomaly from Kepler's equation
FPA[d]	finds the nominal ABM firing direction for a specified flight path angle
INFIND[a,e]	finds a named area on a disc file
INTFRC[a,e]	converts a number into its integer and fractional parts
INTPTB	computes the position and velocity components by interpolation
LOTMAT[a]	performs triangular decomposition of a matrix
MATADD	performs matrix addition
MATMUL	performs matrix multiplication
MODAT[a]	computes atmospheric density
PDAUXP[a,f]	auxiliary for DEQRSPT, computes perturbing accelerations
OFPA[b]	optimises the flight path angle and finds the corresponding nominal ABM firing direction
RANDOMNO[g]	generates a random number in the range 0 to 1.
RLEASE[g]	releases peripheral unit
SCPROD[a,e]	forms scalar product
SIDANG	calculates the sidereal angle at a given epoch
SMPOS[a]	reads sun/moon coordinates from disc and interpolates
STATIS	produces statistical information
TRINV[a]	determines polar coordinates from cartesians (two-dimensional)
TRNFRM	performs the transformation $\underline{X} = \underline{A} + \underline{B}\ \underline{C}$
XROT	rotates rectangular axes through a given angle
UTD4[g]	permits disc core transfers.

(a) These subprograms are common to both POINT2 and SYNMAP. Specifications for them are given in Appendix B.

(b) These subroutines appear in the subprogram that calls them as STAC.

(c) These subroutines are copies of ARANOR for which a specification is given in Appendix B. Only the names have been changed. Groups of six random normal deviates are only required for runs which take tracking errors into account. Separate subroutines are therefore provided so that the same pairs of random normal deviates will be generated irrespective of whether tracking errors are to be included or not.

(d) These subroutines appear in the subprogram that calls them as DUM.

(e) These subprograms, written in PLAN, are provided as semi-compiled segments.

(f) PDAUXP appears in the subprogram that calls it as DAUX.

(g) Provided automatically by 1900 series compilers and not described here.

Appendix D

SYNMAP SUBPROGRAM SPECIFICATIONS

(Specifications for subprograms common to POINT2 and SYNMAP are given in Appendix B only.)

SUBROUTINE ANST

Summary - The subroutine calculates the delta-velocity required for a geosynchronous satellite to acquire station by the shortest path after ABM burn.

Language - 1900 Fortran

Author - M.D. Palmer (August 1975)

Subroutine statement - SUBROUTINE ANST(NO)

Input argument -

NO The case number. If NO = 0, no line printer output is produced.

Use of COMMON - Certain quantities in the common blocks /CNST/, /STATION/ and /STATS/ are used as follows:-

Input argument in /CNST/ -

ENO — Earth's rotation rate (rad/d).

RAD — Conversion factor, degrees to radians.

Input arguments in /STATION/ -

X, Y, Z — Satellite's geocentric position components, $\underline{r}_b = (x, y, z)$ at ABM burn (km).

XDOT, YDOT, ZDOT — Satellite's geocentric velocity components at $\underline{v}_d = (\dot{x}, \dot{y}, \dot{z})$ immediately after ABM burn (km s^{-1}).

MJDB — Modified Julian day number corresponding to the ABM burn time.

TB — Time of ABM burn, in fractions of a day, relative to MJDB.

TR — The time allowed for tracking and orbit determination after ABM burn (days).

C1 — C1 + TR is the total time available for station acquisition (days).

PHI — Station longitude (rad E of Greenwich).

XIL, XIH — Minimum and maximum permitted values of drift orbit inclination (rad).

Input arguments in /STATS/ -

RAS, DEC	Right ascension and declination of the actual ABM firing direction (rad)

Output arguments in /STATS/ -

DV	Total delta-velocity required for station acquisition ($m\ s^{-1}$).
XINCD	Drift orbit inclination (degree).
BO	Right ascension of the drift orbit ascending node (degree).
XLOND	Satellite's longitude at the ABM burn point (^oE).
A	Drift orbit semi major axis (km).
E	Drift orbit eccentricity.

Source deck - 50 cards (ICL code).

Local storage used - 2 real array elements; 21 real variables, and 1 integer variable.

Subordinate subprograms - The subroutines COTOEL and TRINV and the functions ANGLE, EAFKEP and SIDANG.

Explanation - The subroutine uses linearised equations to calculate the total delta-velocity required for a synchronous orbit satellite to acquire station by the shortest path from the ABM burn point. The total delta-velocity required for this strategy is not necessarily a minimum.

The value of the earth's gravitational constant is taken to be 398616.82 km^3/s^2 which takes into account the effect of the second zonal harmonic J_2.

COTOEL is called to find the osculating elements of the drift orbit immediately after ABM burn. The inclination is compared with the maximum and minimum permitted values, and if necessary, the delta-velocity required to place the inclination within these bounds is calculated. The direction in which the satellite is to be moved is identified and the necessary drift rate computed. If the actual drift rate is inadequate or in the wrong direction, a manoeuvre is performed. It is assumed that this and the final stopping manoeuvre are also used to reduce the orbital eccentricity. Finally, the total velocity increment needed to remove any residual eccentricity is calculated.

If NO $\neq$ 0, two lines of information comprising the case number, the ABM firing attitude, drift orbit elements, longitude of the ABM burn point and the required delta-velocity components are output to the line printer.

SUBROUTINE APMAN

Summary - The subroutine simulates the firing of the apogee boost motor (ABM) during the launch phase of a synchronous orbit mission.

Language - 1900 Fortran.

Author - M.D. Palmer (April 1975).

Subroutine statement - SUBROUTINE APMAN (DUM, EP, EPS, GAM).

Input arguments -

DUM — The name of a subordinate subroutine which determines the ABM firing direction. Depending on the chosen ABM firing strategy it may be:-

FPA for a preset flight path angle,
OFPA for an optimised flight path angle,
DR for a preset drift rate.

EP — Time of the epoch to which the random orbits relate, in fractions of a day, relative to MJDOCH (days).

EPS — Epoch time, in seconds relative to MJDOCH (*ie* EPS = 86400 × EP).

GAM — If DUM ≡ FPA, GAM is the required flight path angle (rad).

If DUM ≡ OFPA, GAM is set to 1000.0 and used as a flag.

If DUM ≡ DR, GAM is the required drift rate (rad s^{-1}).

Use of COMMON - Certain quantities in the common blocks /ABM/, /CINTEG/, /CNST/, /CONST/, /CSMOON/ and /STATION/ are used as follows:-

Input arguments in /ABM/ -

HX, HY, HZ — Direction cosines of the unit vector normal to the required drift orbit plane.

DELTAV — Nominal ABM velocity increment (km s^{-1})

DRA — Desired right ascension of the drift orbit (rad)

SDIN — $\sin i_d$, where i_d is the desired drift orbit inclination.

Input arguments in /CINTEG/ -

MJDOCH	Modified Julian day number of the epoch to which the random orbits relate.
P(3)	Satellite's geocentric position components, $\underline{r} = (x, y, z)$, at the latest integration step (km).
V(3)	Satellite's geocentric velocity components, $\underline{v} = (\dot{x}, \dot{y}, \dot{z})$, at the latest integration step (km s^{-1}).
A(3)	Satellite's geocentric acceleration components, $\dot{\underline{v}} = (\ddot{x}, \ddot{y}, \ddot{z})$, at the latest integration step (km s^{-2}).

Input arguments in /CNST/ -

RAD	Conversion factor, degrees to radians.

Input arguments in /CONST/ -

EMU	Earth's gravitational constant (km^3 s^{-2}).

Input arguments in /CSMOON/ -

MJDT	Modified Julian day number of the current time.
TIMET	The current time in fractions of a day relative to MJDT.

Input arguments in /STATION/ -

SUB	The name of the subordinate subroutine to be used to determine the velocity increment required for station acquisition. (Only required if DUM ≡ OFPA.)

Output arguments in /ABM/ -

PT(3)	Satellite's geocentric position components, $\underline{r}_b = (x, y, z)$, at the ABM burn point (km).
VT(3)	Satellite's geocentric velocity components, $\underline{v}_b = (\dot{x}, \dot{y}, \dot{z})$, at the ABM burn point (km/s).
RB	Satellite's radial distance, r_b, at the ABM burn point (km).
VB	Satellite's transfer orbit velocity, v_b, at the ABM burn point (km/s).

Output arguments in /CINTEG/ -

XVEL, YVEL, ZVEL	The initial values of dx/ds, dy/ds and dz/ds.
TVEL	The initial value of dt/ds (s^{-1}).

Output arguments in /STATION/ -

MJDB	Modified Julian day number corresponding to the ABM burn time.
TB	Time of the ABM burn, in fractions of a day, relative to MJDB.

Source deck - 66 cards (ICL code).

Local storage used - 8 real variables; 9 real array elements and 1 integer variable.

Subordinate subprograms - The subroutines ANST, COTOEL, DEQRSPT, DR, DVST, FPA, INTPTB, MODAT, OFPA, PDAUXP, SMPOS, TRINV and the functions ANGLE, ANGL1, EAFKEP, INFIND, INTFRC, SCPROD and SIDANG.

Explanation - The subroutine is provided with the satellite's geocentric position and velocity components at a point on the transfer orbit some 2½ hours before the apogee selected for ABM burn. The orbit is integrated forward to the point at which the transfer and drift orbit planes intersect. At this point, the value of $\underline{r} \cdot \hat{\underline{h}}$, where $\underline{r}$ is the radius vector and $\hat{\underline{h}}$ is the unit vector normal to the required drift orbit plane, changes sign. Thus the integration steps before and after the intersection can be identified. The time of intersection (*ie* the ABM firing time) is found by interpolation. A check is made that the ABM velocity increment (δv) is adequate, *ie* $\underline{v}_b \cdot \hat{\underline{h}} < \delta v$. If not, an error message is printed and a STOP VB.H instruction obeyed.

Depending on the firing strategy specified, a call is made to one of the subroutines DR, FPA or OFPA to find the nominal ABM pointing direction. The name of the subroutine to be used appears as an argument in the call statement.

BLOCK DATA (SYNMAP)

Summary - The Fortran BLOCK DATA segment is used to set initial values for certain quantities stored in common blocks /CINTEG/, /CNST/, /CON/, /CONST/ and /PETURB/.

Language - 1900 Fortran.

Author - M.D. Palmer (April 1975).

Data in /CINTEG/

Variable name	Value	Explanation
I(2)	4	The number of equations to be integrated
H	π/48	Integration step length

Data in /CNST/

Variable name	Value	Explanation
PI	3.1415926536	π
PI2	1.5707963268	π/2
ENO	6.300387476	Earth's rotation rate (rad/day)
RAD	0.0174532925	π/180
SR	42164.75	Synchronous orbit radius (km)
THEL	65.0	Lowest and highest permitted values
THEH	115.0	of solar aspect angle (degree)

Data in /CON/

Variable name	Value	Explanation
EJ5	-0.246E-6	Earth's fifth zonal harmonic J_5
EJ6	0.558E-6	Earth's sixth zonal harmonic J_6
EJ7	-0.326E-6	Earth's seventh zonal harmonic J_7
EJ8	-0.209E-6	Earth's eighth zonal harmonic J_8
EJ9	-0.094E-6	Earth's ninth zonal harmonic J_9
C43	1.0390E-6	Tesseral harmonic coefficient C_{43}
S43	-0.1192E-6	Tesseral harmonic coefficient S_{43}
C41	-0.5175E-6	Tesseral harmonic coefficient C_{41}
S41	-0.4814E-6	Tesseral harmonic coefficient S_{41}
C32	0.7783E-6	Tesseral harmonic coefficient C_{32}
S32	-0.7552E-6	Tesseral harmonic coefficient S_{32}

Data in /CONST/ -

Variable name	Value	Explanation
EMU	398601.3 $km^3\ s^{-2}$	Earth's gravitational constant
EJ2	1082.637E-6	Earth's second zonal harmonic J_2
EJ3	-2.5310E-6	Earth's third zonal harmonic J_3
EJ4	-1.6190E-6	Earth's fourth zonal harmonic J_4
C22	2.4369E-6	Tesseral harmonic coefficient C_{22}
S22	-1.4005E-6	Tesseral harmonic coefficient S_{22}
C33	0.7387E-6	Tesseral harmonic coefficient C_{33}
S33	1.4343E-6	Tesseral harmonic coefficient S_{33}
C44	-0.1846E-6	Tesseral harmonic coefficient C_{44}
S44	0.2508E-6	Tesseral harmonic coefficient S_{44}
C31	2.0192E-6	Tesseral harmonic coefficient C_{31}
S31	0.2278E-6	Tesseral harmonic coefficient S_{31}
C42	0.3444E-6	Tesseral harmonic coefficient C_{42}
S42	0.7021E-6	Tesseral harmonic coefficient S_{42}
ERAD	6378.163 km	Earth's mean equatorial radius
EOMEGA	7.292115147E-5 rad s^{-1}	Earth's rotation rate
PREC	6.079E-12 rad s^{-1}	Earth's precession rate
SUNMU	1.327127E+11 $km^3\ s^{-2}$	Sun's gravitational constant
SELMU	4902.756 $km^3\ s^{-2}$	Moon's gravitational constant

Data in /PERTURB/ -

Variable name	Value	Explanation
HALFCD	1.1	Product $\frac{1}{2}C_D$ where C_D is the drag coefficient.
RATE	1.1	Ratio of the angular velocity of the atmosphere to that of the earth.

SUBROUTINE COTOEL

Summary - The subroutine derives the standard elliptic (osculating) orbital elements of an earth satellite, given its position and velocity components.

Language - ASA Fortran (Standard Fortran 4).

Author - R.H. Gooding (May 1976).

Subroutine statement - SUBROUTINE COTOEL (X, Y, Z, XDOT, YDOT, ZDOT, EMU, A, E, ORBINC, RANODE, ARGPER, EM, EN).

Input arguments -

X, Y, Z	Geocentric cartesian components, (x,y,z), of the satellite, x towards the vernal equinox and z towards the north pole.
XDOT, YDOT, ZDOT	Velocity of the satellite, $(\dot{x},\dot{y},\dot{z})$, measured in the same coordinate system.
EMU	Earth's gravitational constant, μ .

Output arguments -

A	Semi-major axis, a , in the same units as x, y, z
E	Eccentricity, e .
ORBINC	Orbital inclination, i .
RANODE	Right ascension of the ascending node, Ω .
ARGPER	Argument of perigee, ω .
EM	Mean anomaly, M .
EN	Mean motion, n .

Use of COMMON - None.

Source deck - 30 cards, including four comment cards (ICL code).

Local storage used - 13 real variables.

Subordinate subprograms - The subroutine TRINV.

Explanation - Although the subroutine essentially transforms from the six components of the position and velocity of a satellite to its six orbital elements, a seventh input argument permits an arbitrary value of the constant μ

to be used; the seventh output argument, n, is derived from a and μ by the relation $n^2a^3 = \mu$ (Kepler's third law). This means that the subroutine may be used very generally; e.g. for a planet, taking the sun's μ and interpreting Ω as celestial longitude. Units of time and distance are arbitrary but must, of course, be consistent; all angles are in radians.

The subroutine has been written to give maximum accuracy. All angles are derived from knowledge of both sine and cosine, and in such an order that there is no difficulty near the singularities at $e = 0$, $i = 0$ and $i = \pi$. The general method is that of Brouwer and Clemence[15] (section 27 of chapter 1).

The first quantities to be derived are a, e and the eccentric anomaly, E. (NB The Fortran variable E refers to the eccentricity and not the eccentric anomaly.) These come from the relations

$$\frac{1}{a} = \frac{2}{r} - \frac{V^2}{\mu} ,$$

$$e \cos E = \frac{rV^2}{\mu} - 1$$

and

$$e \sin E = \frac{(x\dot{x} + y\dot{y} + z\dot{z})}{(\mu a)^{\frac{1}{2}}} ,$$

where $r = (x^2 + y^2 + z^2)^{\frac{1}{2}}$ and $V^2 = \dot{x}^2 + \dot{y}^2 + \dot{z}^2$.

Then M follows at once, since

$$M = E - e \sin E .$$

Note that if e is close to zero E is ill-determined, reflecting the indeterminancy of perigee - and in fact if $e = 0$, E is set to 0. This does not matter at all, since whatever position of perigee is specified by the value taken for E, the value of M is fully consistent with it.

The values of i, Ω and ω are derived from the basic matrix relation

$$\begin{pmatrix} \cos\omega & \sin\omega & 0 \\ -\sin\omega & \cos\omega & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos i & \sin i \\ 0 & -\sin i & \cos i \end{pmatrix} \begin{pmatrix} \cos\Omega & \sin\Omega & 0 \\ -\sin\Omega & \cos\Omega & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} P_x & P_y & P_z \\ Q_x & Q_y & Q_z \\ R_x & R_y & R_z \end{pmatrix} ,$$

where $P_x = (x/r)\cos E - \dot{x}(a/\mu)^{\frac{1}{2}} \sin E$,

$Q_x = (1 - e^2)^{-\frac{1}{2}} [(x/r)\sin E + \dot{x}(a/\mu)^{\frac{1}{2}}(\cos E - e)]$,

$R_x = (\mu a(1 - e^2))^{-\frac{1}{2}}(y\dot{z} - z\dot{y})$,

and similarly for P_y, P_z, Q_y, Q_z, R_y and R_z .

Thus i and Ω are derived from

$$\sin i \sin \Omega = R_x ,$$

$$-\sin i \cos \Omega = R_y ,$$

and

$$\cos i = R_z ,$$

with an indeterminacy in Ω when i is close to 0 or π (Ω is set to 0 if $\sin i = 0$).

Finally ω is determined from

$$\sin i \cos \omega = Q_z$$

and

$$\sin i \sin \omega = P_z$$

where we have to be sure that no difficulty arises over the e-singularity or either of the i-singularities.

There is no trouble near the e-singularity since the quantities $e \cos E$ and $e \sin E$ are used (through Q_z and P_z), in the derivation of ω , in the same ratio as in the derivation of E , so that $E + \omega$ is always correct. (When E is set conventionally to 0 because $e = 0$, $e \cos E$ is set to 1 (!) to ensure the correct ratio between Q_z and P_z .)

In the same way, ω has to be compatible with Ω near the i-singularity, and this is automatic when z and $\dot{z}$ are not both exactly zero, since Ω and ω both depend on the ratio of these two quantities. When z and $\dot{z}$ *are* both exactly zero, the correct value of ω is achieved by replacing P_z and Q_z by P_y and Q_y , with an additional factor to ensure that ω is in the correct quadrant.

SUBROUTINE DR

Summary - The subroutine computes the direction cosines of the nominal ABM firing direction required to obtain a specified drift rate after ABM burn.

Language - 1900 Fortran.

Author - M.D. Palmer (May 1975).

Subroutine statement - SUBROUTINE DR (AD)

Input arguments -

AD — The drift orbit semi major axis corresponding to the required drift rate (km).

Use of COMMON - Certain arguments in the common blocks /ABM/, /CNST/ and /CONST/ are used as follows:-

Input arguments in /ABM/ -

PT(3), VT(3) — The satellite's transfer orbit geocentric position, $\underline{r}_b = (x, y, z)$, and velocity, $\underline{v}_b = (\dot{x}, \dot{y}, \dot{z})$, components at the ABM burn point (km, km s^{-1}).

R, VB — The satellite's radial distance, r_b, and velocity, v_b, at the ABM burn point (km, km s^{-1}).

HX, HY, HZ — Direction cosines of the unit vector, $\hat{\underline{h}}$, normal to the required drift orbit plane.

DV — Nominal ABM velocity increment (km s^{-1}).

Input argument in /CNST/ -

PI — π .

PI2 — $\pi/2$.

Input argument in /CONST/ -

EMU — Earth's gravitational constant (km^3 s^{-2}).

Output arguments in /ABM/ -

SX, SY, SZ — Direction cosines of the required ABM firing direction.

Source deck - 56 cards. (ICL code).

Local storage used - 37 real variables.

Subordinate subprograms - None.

Explanation - The subroutine calculates the direction cosines of the ABM firing direction required to produce a specified drift rate after ABM burn. This parameter is passed to the subroutine as the equivalent drift orbit semi major axis.

The drift orbit velocity required after ABM burn is calculated together with the angle (θ) between the transfer and drift orbit velocity vectors and the angle (ϕ) between the transfer orbit velocity vector and the unit vector $\underline{\hat{h}}$. If the ABM velocity increment is inadequate, *ie* $\phi + \theta < 90^{\circ}$, a STOP ABM DV instruction is executed.

Let the direction cosines of the drift orbit velocity vector, the unit vector $\underline{\hat{h}}$ and the transfer orbit velocity vector be (ℓ_x, ℓ_y, ℓ_z), (h_x, h_y, h_z) and (b_x, b_y, b_z) respectively. Then

$$b_x\ell_x + b_y\ell_y + b_z\ell_z = \cos\theta ,$$

$$\ell_x^2 + \ell_y^2 + \ell_z^2 = 1 ,$$

and

$$h_x\ell_x + h_y\ell_y + h_z\ell_z = 0 .$$

Therefore

$$\ell_z = -[h_x\ell_x + h_y\ell_y]/h_z$$

and

$$b_x\ell_x + b_y\ell_y - \frac{b_z[h_x\ell_x + h_y\ell_y]}{h_z} = \cos\theta$$

ie

$$\left[b_x - \frac{b_z h_x}{h_z}\right]\ell_x + \left[b_y - \frac{b_z h_y}{h_z}\right]\ell_y = \cos\theta$$

or

$$A\ell_x + B\ell_y = \cos\theta .$$

Setting

$$\ell_x = [\cos\theta - B\ell_y]/A ,$$

we have

$$\ell_x^2 + \ell_y^2 + [h_x^2\ell_x^2 + 2h_xh_y\ell_x\ell_y + \ell_y^2h_y^2]/h_z^2 = 1$$

ie

$$\left[\frac{\cos\theta}{A} - \frac{B}{A}\ell_y\right]^2\left[1 + \frac{h_x^2}{h_z^2}\right] + 2\,\frac{h_xh_y}{h_z^2}\left[\frac{\cos\theta}{A} - \frac{B}{A}\ell_y\right]\ell_y + \ell_y^2\left[1 + \frac{h_y^2}{h_z^2}\right] = 1$$

or

$$a\ell_y^2 + b\ell_y + c = 0 .$$

If $b^2 - 4ac$ is negative, the program obeys a STOP 90 instruction. If $b^2 > 4ac$ the equation has two possible solutions. These are evaluated and the corresponding values of ℓ_x and ℓ_z determined. The drift orbit eccentricity for each solution is calculated and the direction cosines of the ABM firing direction determined for the solution which gives the minimum drift orbit eccentricity.

SUBROUTINE DVST

Summary	- The subroutine calculates the delta-velocity required for a geosynchronous satellite to acquire station after ABM burn. Both easterly and westerly drifts are examined and the one giving the smaller total delta-velocity selected.
Language	- 1900 Fortran.
Author	- M.D. Palmer (January 1974).
Subroutine statement	- SUBROUTINE DVST(NO)
Input argument	-
NO	The case number. If NO = 0, no line printer output is produced.
Use of COMMON	- Certain quantities in the common blocks /CNST/, /STATION/ and /STATS/ are used as follows:-
Input arguments in /CNST/	-
ENO	Earth's rotation rate (rad/d)
RAD	Conversion factor, degrees to radians.
Input arguments in /STATION/	-
X, Y, Z	Satellite's geocentric position components, $\underline{r}_b = (x, y, z)$, at ABM burn (km).
XDOT, YDOT, ZDOT	Satellite's geocentric velocity components, $\underline{v}_d = (\dot{x}, \dot{y}, \dot{z})$, immediately after ABM burn (km s^{-1}).
MJDB	Modified Julian day number corresponding to the ABM burn time.
TB	Time of ABM burn, in fractions of a day, relative to MJDB.
TR	The time allowed for tracking and orbit determination after ABM burn (days).
C1	C1 + TR is the total time available for station acquisition (days).
PHI	Station longitude (rads E of Greenwich).

XIL, XIH — Minimum and maximum permitted values of drift orbit inclination (rad).

Input arguments in /STATS/ -

RAS, DEC — Right ascension and declination of the actual ABM firing direction (rad).

Output arguments in /STATS/ -

DV — Total delta-velocity required for station acquisition ($m\ s^{-1}$).

XINCD — Drift orbit inclination (degree).

BO — Right ascension of the ascending node of the drift orbit (degree).

XLOND — Satellite's longitude at the ABM burn point (degree E).

A — Drift orbit semi major axis (km).

E — Drift orbit eccentricity.

Source deck - 49 cards (ICL code).

Local storage used - 4 real array elements; 20 real variables and 1 integer variable.

Subordinate subprograms - The subroutines COTOEL and TRINV and the functions ANGLE, EAFKEP and SIDANG.

Explanation - The subroutine uses linearised equations to calculate the total delta-velocity required for a synchronous satellite to acquire station. Both easterly and westerly drifts are examined and the one requiring the smaller delta-velocity selected.

The value of the earth's gravitational constant is taken to be $398616.82\ km^3\ s^{-2}$ which takes into account the effects of the earth's second zonal harmonic J_2.

COTOEL is called to find the osculating elements of the drift orbit immediately after ABM burn. The satellite longitude at the ABM burn point is calculated. If $NO \neq 0$, the case number, the ABM firing attitude, drift orbit elements and the longitude of the ABM burn point are output to the lineprinter. The inclination is compared with the maximum and minimum permitted values and, if necessary, the delta-velocity required to place the inclination within these bounds is determined.

Both easterly and westerly drifts are now examined. In each case, the necessary drift rate is computed. If the actual drift rate is inadequate or in the wrong direction a manoeuvre is performed. It is assumed that this and the final stopping manoeuvre are also used to reduce the orbital eccentricity. Finally the total velocity increment needed to remove any residual eccentricity is calculated. If NO $\neq 0$, one line is printed giving, for each direction, the drift rate and the velocity increments required.

FUNCTION EAFKEP

Summary - The *function solves* Kepler's *equation; i.e. it provides* the eccentric anomaly of a celestial body, E, given the orbital eccentricity, e, and mean anomaly, M. Kepler's equation is $M = E - e \sin E$.

Language - ASA Fortran (Standard Fortran 4).

Author - R.H. Gooding (May 1976).

Function statement - FUNCTION EAFKEP (EM, ECC).

Input arguments -

EM Mean anomaly, M (radians).

ECC Eccentricity, e.

Output function -

EAFKEP Eccentric anomaly, E.

Use of COMMON - None.

Source deck - 13 cards, including 3 comment cards (ICL code).

Local storage used - 3 real variables.

Subordinate subprograms - None.

Explanation - A first approximation to E is given by $E_1 = M$. Improved approximations are given by Newton's method; thus

$$E_{i+1} = E_i + (M - E_i + e \sin E_i)\big/(1 - e \cos E_i) .$$

The process ends after three iterations if $e < 0.003$, and otherwise after five. This ensures sufficient accuracy at all times, while making the subroutine independent of the word-length of the computer. (If the magnitude of $|E_{i+1} - E_i|$ were used as a criterion for convergence, the numerical value it was compared with would have to vary from computer to computer.)

SUBROUTINE FPA

Summary - The subroutine finds the direction cosines of the nominal apogee boost motor (ABM) firing direction required to obtain a specified flight path angle at ABM burn.

Language - 1900 Fortran.

Author - M.D. Palmer (May 1975).

Subroutine statement - SUBROUTINE FPA (GAM)

Input argument -

GAM — The required flight path angle, γ (rad).

Use of COMMON - Certain quantities in common block /ABM/ are used as follows:-

Input arguments in /ABM/ -

PT(3), VT(3) — The satellite's transfer orbit geocentric position, $\underline{r}_b = (x, y, z)$, and velocity, $\underline{v}_b = (\dot{x}, \dot{y}, \dot{z})$, components at the ABM burn point (km, km s^{-1}).

RB, VB — The satellite's radial distance and velocity at the ABM burn point (km, km s^{-1}).

HX, HY, HZ — Direction cosines of the unit vector ($\hat{\underline{h}}$) normal to the required drift orbit plane.

DELTAV — Nominal ABM velocity increment (km s^{-1}).

Output arguments -

SX, SY, SZ — Direction cosines of the required ABM firing direction.

Source deck - 24 cards (ICL code).

Local storage used - 12 real variables.

Subordinate subprograms - None.

Explanation - Let $\hat{\underline{r}}_b$ be a unit vector along the radius vector at the ABM burn point and define the unit vector $\hat{\underline{u}}$ such that $\hat{\underline{u}}, \hat{\underline{h}}, \hat{\underline{r}}_b$ form a right handed orthogonal set (see Fig 5). Then

$$\hat{\underline{u}} = \hat{\underline{r}}_b \times \hat{\underline{h}} .$$

The flight path angle (γ) is measured from $\hat{\underline{u}}$ in the plane defined by $\hat{\underline{u}}$ and $\hat{\underline{r}}_b$. If $\hat{\underline{v}}_d$ is a unit vector along the drift orbit velocity vector $\underline{v}_d$, then

$$\hat{\underline{v}}_d = \hat{\underline{u}} \cos\gamma + \hat{\underline{r}}_b \sin\gamma$$

and

$$|\underline{v}_d| = |\underline{v}_b \cdot \hat{\underline{v}}_d| + [\delta v^2 - (v_b^2 - |\underline{v}_b \cdot \hat{\underline{v}}_d|^2)]^{\frac{1}{2}} ,$$

where δv is the nominal ABM velocity increment. If the argument of the square root is negative, a STOP FPA instruction is executed. The direction cosines of the nominal ABM firing direction, (s_x, s_y, s_z), are found from the equation

$$s_x = \Big[|v_d|\ell_x - |v_b|b_x\Big]/\delta v ,$$

$$s_y = \Big[|v_d|\ell_y - |v_b|b_y\Big]/\delta v$$

and

$$s_z = \Big[|v_d|\ell_z - |v_b|b_z\Big]/\delta v$$

where (ℓ_x, ℓ_y, ℓ_z) are the direction cosines of the drift orbit velocity vector and (b_x, b_y, b_z) are the direction cosines of the transfer orbit velocity vector.

SUBROUTINE INTPTB

Summary - The subroutine calculates the cartesian components of the geocentric position and velocity of an earth satellite at a given time by interpolation between two points in the orbit. These points are defined by position, velocity and acceleration components at specified times.

Language - 1900 Fortran.

Author - M.D. Palmer (July 1974)

Subroutine statement - SUBROUTINE INTPTB (TINT, TP, TIMET, PPVA, PT, VT).

Input arguments -

TINT	Time, t , at which the interpolation is required.
TP	Time, t_1 , of the point prior to the interpolation.
TIMET	Time, t_2 , of the point after the interpolation.
PPVA(9)	Satellite's geocentric position, velocity and acceleration components at time t_1 (x_1, y_1, z_1 etc.).

Output arguments -

PT(3)	Satellite's geocentric position components at time t (x, y, z).
VT(3)	Satellite's geocentric velocity components at time t ($\dot{x}, \dot{y}, \dot{z}$).

Use of COMMON - Certain arguments in the common block /CINTEG/ are used as follows:

Input arguments -

PP(3)	Satellite's geocentric position components at time t_2 (x_2, y_2, z_2).
V(3)	Satellite's geocentric velocity components at time t_2 ($\dot{x}_2, \dot{y}_2, \dot{z}_2$).
XACCT(3)	Satellite's geocentric acceleration components at time t_2 ($\ddot{x}_2, \ddot{y}_2, \ddot{z}_2$).

<u>Local storage used</u> - 23 real variables and 1 integer variable.

<u>Subordinate subprograms</u> - None.

<u>Source deck</u> - 40 cards (ICL code).

<u>Explanation</u> - The subroutine obtains a satellite's geocentric position and velocity components at a given time by interpolation between two points in the orbit. These will normally be adjacent steps in a numerical integration.

These equations used for interpolation are:

$$f(t) = q^3\left[(1 + 3p + 6p^2)f(t_1) + hp(1 + 3p)f'(t_1) + \frac{h^2}{2}p^2 f''(t_1)\right] + p^3\left[(1 + 3q + 6q^2)f(t_2) - hq(1 + 3q)f'(t_2) + \frac{h^2}{2}q^2 f''(t_2)\right]$$

for position components and

$$f'(t) = 30h^{-1}p^2q^2\left[f(t_2) - f(t_1)\right] + q^2(1 + 5p)(1 - 3p)f'(t_1) + p^2(1 + 5q)(1 - 3q)f'(t_2) + \frac{h}{2}pq^2(2 - 5p)f''(t_1) - \frac{h}{2}p^2q(2 - 5q)f''(t_2)$$

for velocity components, where $h = t_2 - t_1$, $p = (t - t_1)/h$, $q = 1 - p$ and $f(t_i)$, $f'(t_i)$ and $f''(t_i)$ are the position, velocity and acceleration components at time t_i .

SUBROUTINE MATADD

Summary - The subroutine adds two matrices of order $m \times n$.

Language - 1900 Fortran.

Author - M.D. Palmer (July 1974)

Subroutine statement - SUBROUTINE MATADD (A, B, C, M, N).

Input arguments

$\left.\begin{array}{l}A\\B\end{array}\right\}$ Matrices of order $M \times N$.

M Number of rows.

N Number of columns.

Output arguments

C Matrix of order $M \times N$ $(C = A + B)$.

Use of COMMON - None

Source deck - 8 cards (ICL code).

Local storage used - 2 integer variables.

Subordinate subprograms - None.

Explanation - The subroutine performs the matrix operation $C = A + B$. Each element of C is obtained by the addition of the corresponding elements of A and B .

SUBROUTINE MATMUL

Summary	- The subroutine performs matrix multiplications.
Language	- USA Standard Fortran (USAS X3.9 - 1966).
Author	- G.E. Cook (May 1970).
Subroutine statement	- SUBROUTINE MATMUL (EM1, EM2, EM3, II, KK, JJ, IA, IB).
Input arguments	-
EM1	Matrix of dimension (II,KK).
EM2	Matrix of dimension (KK,JJ).
II	Actual number of rows in EM1 .
KK	Actual number of columns in EM1 and rows in EM2 .
JJ	Actual number of columns in EM2 .
IA	Maximum number of rows in EM1 as given in the dimension statement of the calling segment.
IB	Maximum number of rows dimensioned for EM2 in the calling segment.
Output arguments	-
EM3	Matrix of dimension (II,JJ).
Use of COMMON	- None.
Source deck	- 13 cards, including 4 comment cards (ICL code).
Local storage used	- 3 integer variables.
Subordinate subprograms	- None.
Explanation	- The subroutine performs the matrix multiplication

$$M_1 M_2 = M_3 \ .$$

SUBROUTINE OFPA

Summary - The subroutine optimises the drift orbit flight path angle at ABM burn such that the total delta-velocity required for station acquisition and circularisation of the drift orbit is a minimum.

Language - 1900 Fortran.

Author - M.D. Palmer (September 1975).

Subroutine statement - SUBROUTINE OFPA (STAC).

Input argument -

STAC The name of the subroutine to be used to compute the delta-velocity required for station acquisition.

Use of COMMON - Certain quantities in common blocks /ABM/, /CNST/, /STATION/ and /STATS/ are used as follows:-

Input arguments in /ABM/ -

PT(3), VT(3) The satellite's geocentric transfer orbit position, $\underline{r}_b = (x, y, z)$, and velocity, $\underline{v}_b = (\dot{x}, \dot{y}, \dot{z})$, components at the ABM burn point (km, km s^{-1}).

RB, VB The satellite's radial distance, r_b , and the transfer orbit velocity, v_b , at the ABM burn point (km, km s^{-1}).

HX, HY, HZ Direction cosines of the unit vector, $\hat{\underline{h}}$, normal to the required drift orbit plane.

DV The nominal ABM velocity increment (km s^{-1})

DRA Right ascension of the ascending node of the required drift orbit (rad).

SDIN Sine i_d , where i_d is the desired drift orbit inclination.

Input argument in /CNST/ -

PI π .

RAD Conversion factor, degrees to radians.

Input argument in /STATS/ -

DVT — Total velocity increment required for station acquisition ($m\ s^{-1}$).

Output arguments in /ABM/ -

S1, S2, S3 — Direction cosines of the required ABM firing direction.

Output arguments in /STATION/ -

X, Y, Z — Satellites geocentric position components at the ABM burn time (km).

V(3) — Satellite's geocentric drift orbit velocity components, $\underline{v}_d = (\dot{x}, \dot{y}, \dot{z})$, immediately after ABM burn ($km\ s^{-1}$).

Source deck - 113 cards (ICL code).

Local storage used - 37 real variables.

Subordinate subprograms - The subroutines ANST, COTOEL, DVST, and TRINV and the functions ANGLE, EAFKEP and SIDANG.

Explanation - The subroutine optimises the drift orbit flight path such that the total velocity increment required for station acquisition is a minimum. The value of the earth's gravitational constant is taken to be 398616.82 $km^3\ s^{-2}$ which takes into account the effects of the earth's second zonal harmonic, J_2 .

The transfer orbit inclination (i_t) and the required nodal rotation ($\Delta\Omega$) are calculated. The angle, $\Delta\alpha$, between the transfer and drift orbit planes is then obtained using the equation

$$\cos \Delta\alpha = \sin i_d \sin i_t \cos \Delta\Omega + \cos i_d \cos i_t \ ,$$

where i_d is the required drift orbit inclination. The angle ϕ_t between the transfer orbit radius and velocity vectors is given by

$$\cos \phi_t = \hat{\underline{r}}_b \cdot \hat{\underline{v}}_b \ ,$$

where $\hat{\underline{r}}_b$ and $\hat{\underline{v}}_b$ are unit vectors along the transfer orbit radius and velocity vectors respectively.

The angle ψ between the transfer and drift orbit velocity vectors is given by

$$\cos\psi = \cos\phi_t \cos\phi_d + \sin\phi_t \sin\phi_d \cos\Delta\alpha \quad ,$$

where ϕ_d is the angle between the drift orbit radius and velocity vectors.

If the nominal ABM velocity increment is δv ,

$$\delta v^2 = v_d^2 + v_b^2 - 2v_d v_b \cos\psi \quad ,$$

and so given ψ , v_d can be determined. The subroutine uses an iterative procedure to find the direction cosines (ℓ_x, ℓ_y, ℓ_z) of the drift orbit velocity vector. Let (b_x, b_y, b_z) and (h_x, h_y, h_z) be the direction cosines of the transfer orbit radius vector at the ABM burn point and the normal to the drift orbit plane.

Then

$$\hat{\underline{h}} \cdot \hat{\underline{v}}_d = h_x\ell_x + h_y\ell_y + h_z\ell_z = 0 \quad ,$$

$$\ell_x^2 + \ell_y^2 + \ell_z^2 = 1 \quad ,$$

and

$$\hat{\underline{r}}_b \cdot \hat{\underline{v}}_d = b_x\ell_x + b_y\ell_y + b_z\ell_z = \cos\phi_d \quad .$$

Thus

$$\ell_x = [- h_y\ell_y + h_z\ell_z]/h_x \quad ,$$

and

$$\cos\phi_d = b_y\ell_y + b_z\ell_z - b_x[h_y\ell_y + h_z\ell_z]/h_x \quad ,$$

from which we obtain

$$\ell_y\left[b_y - \frac{b_x h_y}{h_x}\right] + \ell_z\left[b_z - \frac{b_x h_z}{h_x}\right] = \cos\phi_d \quad ,$$

ie

$$A\ell_y + B\ell_z = \cos\phi_d \quad ,$$

or

$$\ell_y = [\cos\phi_d - B\ell_z]/A \quad .$$

It follows that

$$\left[1 + \frac{h_z^2}{h_x^2}\right]\ell_z^2 + \ell_y^2\left[\frac{h_y^2}{h_x^2} + 1\right] + \frac{2h_y h_z \ell_y \ell_z}{h_x^2} = 1$$

and substituting for ℓ_y

$$\left[1 + \frac{h_z^2}{h_x^2}\right] \ell_z^2 + \left[\frac{\cos\phi_d}{A} - \frac{B}{A}\ell_z\right]^2 \left[\frac{h_y^2}{h_x^2} + 1\right] + 2\,\frac{h_y h_z \ell_z}{h_x^2}\left[\frac{\cos\phi_d}{A} - \frac{B}{A}\ell_z\right] = 1 \ .$$

These equations are solved to obtain two values of ℓz and corresponding values of ℓ_y and ℓ_x . The product $\hat{\underline{v}}_d \cdot \hat{\underline{v}}_b$ is formed for each solution and the components of $\underline{v}_d$ are computed for the scalar product which is closest in value to $\cos\psi$. A subroutine is then called to compute the total velocity increment required for station acquisition.

In practice, ϕ_d is set initially to 80 degrees and the delta-velocity computed and stored. ϕ_d is then incremented by $\delta\phi_d$, (4 degrees initially) and the procedure repeated until the delta-velocity passes through a minimum. $\delta\sigma_d$ is then divided by four and the process is repeated until a minimum is found with $\delta\phi_d < 0.001$ radian. The appropriate set of direction cosines are found as above and the direction cosines of the nominal ABM firing direction are obtained using the equation

$$\delta vs_1 = |\underline{v}_d|\ell_x - v_{b_x} \ ,$$

$$\delta vs_2 = |\underline{v}_d|\ell_y - v_{b_y} \ ,$$

and

$$\delta vs_3 = |\underline{v}_d|\ell_z - v_{b_z} \ .$$

FUNCTION SIDANG

Summary - The function calculates the modified sidereal angle corresponding to a given date/time (t).

Language - A.S.A. Fortran (Standard Fortran 4).

Author - R.J. Tayler (September 1969)

Function statement - FUNCTION SIDANG (MJD, TIME).

Input arguments -

MJD Modified Julian day number

TIME Time (fraction of a day) such that t is given by MJD + TIME.

Output function -

SIDANG The modified sidereal angle (radians)

Use of COMMON - None.

Source deck - 11 cards, including 5 comment cards (ICL code).

Local storage used - None.

Subordinate subprograms - None.

Explanation - The modified sidereal angle is a quantity which differs from the more familiar sidereal time only because of the choice of a non-standard reference direction. This reference direction lies in the plane of the true equator and is given by rotation of the true equinox, by an amount equal to the precession and nutation in right ascension since 1950.0 but in the opposite sense.

In *degrees*, the formula for modified sidereal angle is

$$\hat{\theta} = 100^{\circ}.075542 + 360^{\circ}.985612288\ (d - 33282.0)$$

ie the origin has been adjusted slightly from 1950.0, which occurred at MJD 33281.9234. However, SIDANG operates by calculating the angle in revolutions, as

$$0.277987616 + 0.00273781191\ (\mathrm{MJD} - 33282) + 1.00273781191\ \mathrm{TIME}$$

with an integral number of revolutions already dropped, and then converting to *radians* (after dropping any remaining full revolutions) by multiplying by 6.2831853072.

SUBROUTINE STATIS

Summary	-	The subroutine produces statistical information about the orbital elements of the drift orbit and the delta-velocity required for station acquisition.
Language	-	1900 Fortran.
Author	-	M.D. Palmer (April 1976).
Subroutine statement	-	SUBROUTINE STATIS (RNC)
Input argument	-	
RNC		On the first n calls, n being the number of cases, RNC = 0 . On the n + 1 th call, RNC = 1/n. (see explanation).
Use of COMMON		Certain quantities in common block /STATS/ are used as follows:-
Input arguments in /STATS/	-	
RA, DEC		Right ascension and declination of the ABM firing direction (degree).
DV		Delta-velocity required for station acquisition (m/s).
OI		Drift orbit inclination (degree).
BO		Right ascension of the ascending node of the drift orbit (degree).
XLON		Satellite longitude at ABM burn (degrees east of Greenwich).
A		Drift orbit semi major axis (km).
E		Drift orbit eccentricity.
THEB		Solar aspect angle at ABM burn (degree).
ELEV		Spin axis elevation angle at ABM burn (degree).
NTHEB		The number of cases for which the solar aspect angle constraint has been violated.
Source deck	-	147 cards (ICL code).
Local storage used	-	22 integer variables and 41 real variables.

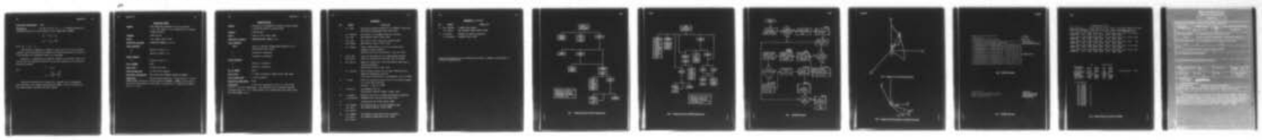
AD-A060 905 ROYAL AIRCRAFT ESTABLISHMENT FARNBOROUGH (ENGLAND) F/G 9/2

THE SYNCHRONOUS MISSION ANALYSIS PROGRAM SYNMAP.(U)

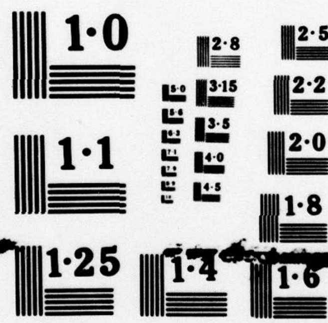
DEC 77 M D PALMER, G E COOK

UNCLASSIFIED RAE-TR-77179 DRIC-BR-64247 NL

2 OF 2
ADA
060905



END
DATE
FILMED
1 -79
DDC



NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

<u>Subordinate subprograms</u> - None.

<u>Explanation</u> - On the ith call (i = 1,...,n) STATIS performs the following operations on each quantity in the common block /STATS/,

$$X_i = X_{i-1} + x_i$$

$$Y_i = Y_{i-1} + x_i^2$$

where $X_0 = Y_0 = 0$.

The satellite longitude is checked to ensure that it is in the correct quadrant, and the delta-velocity required for station acquisition is stored in a form suitable for subsequent output as a histogram.

When the n simulations are complete, STATIS is called with RNC = 1/n and the mean and standard deviations of the above quantities are calculated using the formulae:-

$$\bar{x} = X_n/n$$

and

$$\sigma_x = \left[\frac{Y_n}{n} - \bar{x}^2\right]^{\frac{1}{2}} .$$

This data is output on the lineprinter, together with a histogram of delta-velocities and a comment specifying the number of cases for which the solar aspect angle constraint has been violated.

SUBROUTINE TRNFRM

Summary - The subroutine performs the matrix operation $C = A + BF$ where F is a column vector of random normal deviates.

Language - 1900 Fortran.

Author M.D. Palmer (April 1975).

Subroutine statement - SUBROUTINE TRNFRM (A, B, C).

Input arguments -

A Matrix of order 6×1.

B Matrix of order 6×6.

Output argument -

C Matrix of order 6×1.

Use of COMMON - None.

Source deck - 8 cards (ICL code).

Local storage used - 60 real array elements.

Subordinate subprograms - The subroutines ARANOR6, MATADD and MATMUL.

Explanation - The subroutine generates a column vector F of six random normal deviates by calling subroutine ARANOR6. MATMUL is then used to form BF and this product is then added to the matrix A to complete the operation $C = A + BF$.

SUBROUTINE XROT

Summary - Rotation of a rectangular coordinate system through a specified angle in a single plane.

Language - 1900 Fortran.

Author - Diana W. Scott (April 1969).

Subroutine statement - SUBROUTINE XROT (THETA, Y, Z).

Input arguments -

THETA	Angle (in radians) through which system is to be rotated about the x axis.
Y	Cartesian y coordinate.
Z	Cartesian z coordinate.

Output arguments -

Y	Rotated y coordinate.
Z	Rotated z coordinate.

Use of COMMON - None.

Source deck - 11 cards, including 2 comment cards (ICL code).

Local storage used - 3 real variables.

Subordinate subprograms - None.

Explanation - y and z are replaced by ($y \cos \theta + z \sin \theta$) and ($z \cos \theta - y \sin \theta$) respectively. The subroutine can of course be used for rotations about the y or z axes as well, *eg* to rotate about the y axis, write CALL XROT (THETA, Z, X).

REFERENCES

No.	Author	Title, etc
1	G.E. Cook	Preliminary mission analysis for the SKYNET 3 spacecraft. RAE Technical Memorandum Space 208 (1974)
2	G.J. Davison G.E. Cook	An analysis of the MAROTS injection strategy. RAE Technical Report 76005 (1976)
3	G.E. Cook M.D. Palmer	The elliptic orbit integration program POINT. RAE Technical Report 76129 (1976)
4	W.J. Moonan	Linear transformation to a set of stochastically dependent normal variables. *American Statistical Association J.* 52, 247 (1957)
5	G.E.P. Box M.E. Muller	A note on the generation of random normal deviates. *Annals of Mathematical Statistics,* 29, 610 (1958)
6	R.H. Merson	Numerical integration of the differential equations of celestial mechanics. RAE Technical Report 74184 (1974)
7	L.G. Jacchia	Static diffusion models of the upper atmosphere with empirical temperature profiles. *Smithsonian Contributions to Antrophysics.,* 8, 215 (1965)
8	Y. Kozai	Effect of precession and nutation on the orbital elements of a close earth satellite. *Astronom. J.* 65, 621 (1960)
9	S. Herrick	Antrodynamics (Vol 2). Van Nostrand Reinhold Company, London (1972)
10	L. Lapidus J.H. Seinfeld	Numerical solution of ordinary differential equations. Academic Press, New York and London (1971)
11	-	Interpolation and allied tables, HMSO.
12	P.R. Peabody J.F. Scott E.G. Orozco	User's description of the JPL ephemeris tapes. JPL Technical Report, 32-580 (1964)
13	P.R. Peabody J.F. Scott E.G. Orozco	JPL ephemeris tapes E9510, E9511 and E9512. JPL Technical Memorandum 33-167 (1964)

REFERENCES (concluded)

No.	Author	Title, etc
14	R.H. Gooding R.J. Taylor	A PROP3 users manual. RAE Technical Report 68294 (1968)
15	D. Brouwer G.M. Clemence	Methods of celestial mechanics. Academic Press (1961)

Reports quoted above are not necessarily available to members of the public or commercial organisations.

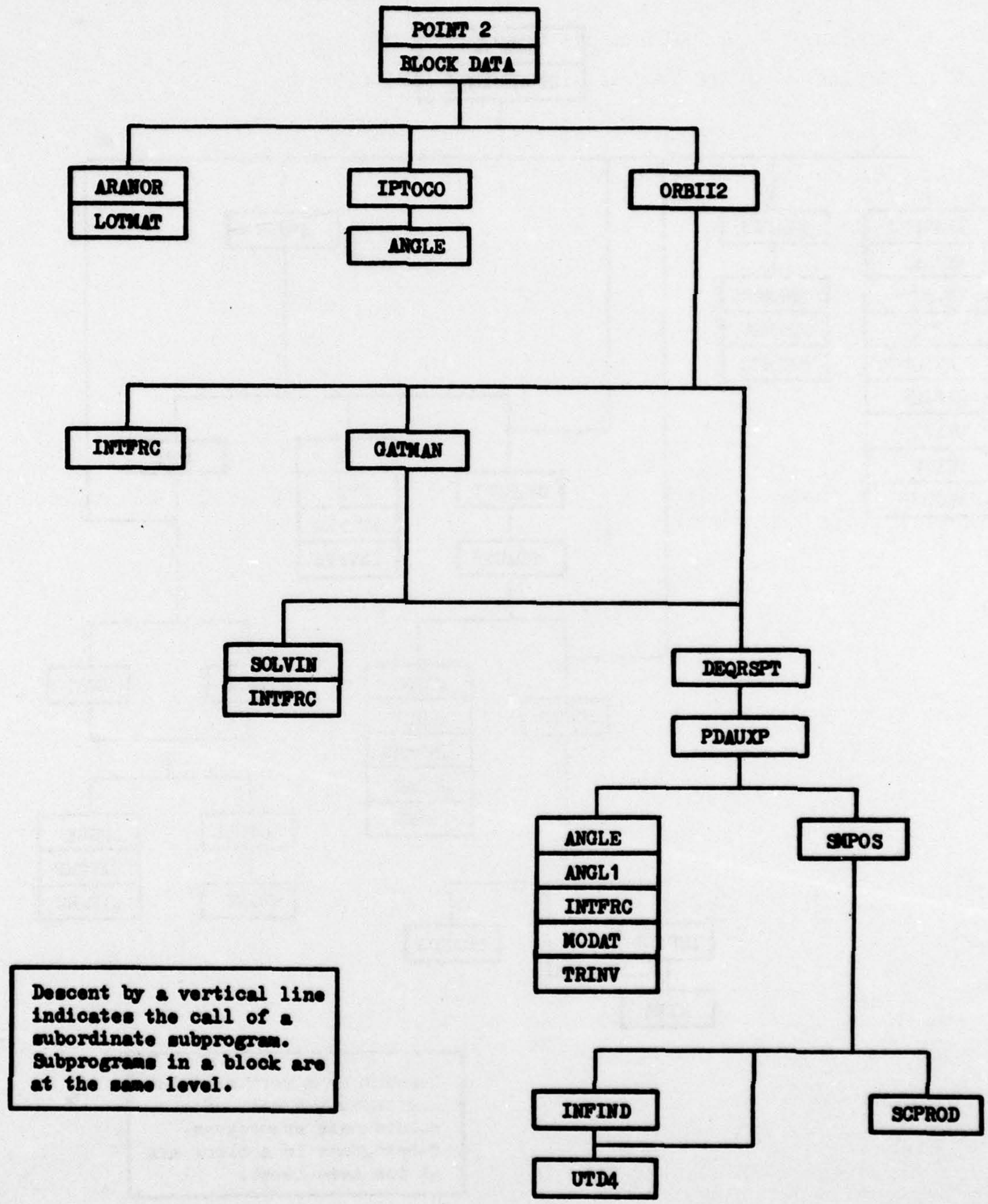


Fig 1 Calling structure for POINT 2 program units

Fig 2

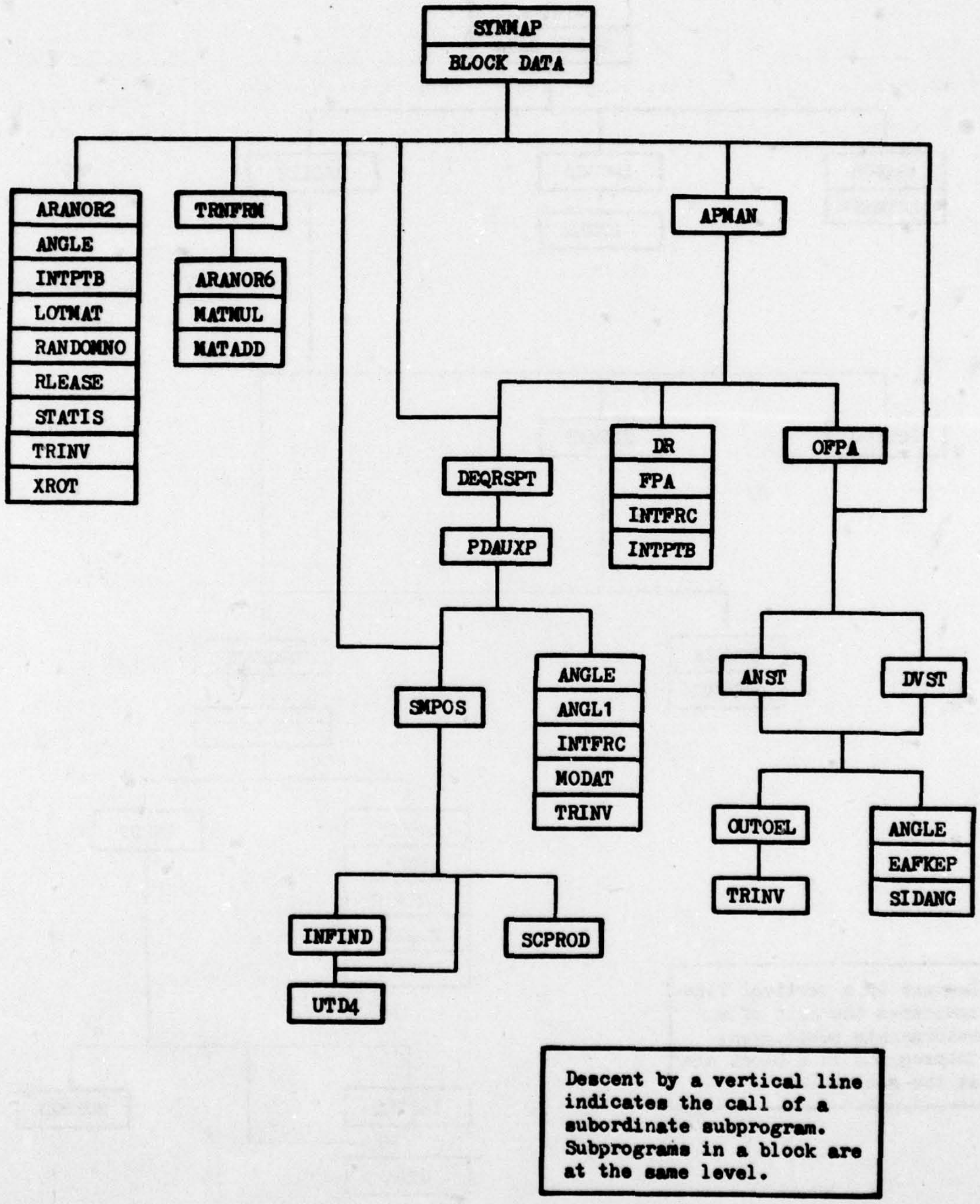


Fig 2 Calling structure for SYNMAP program units

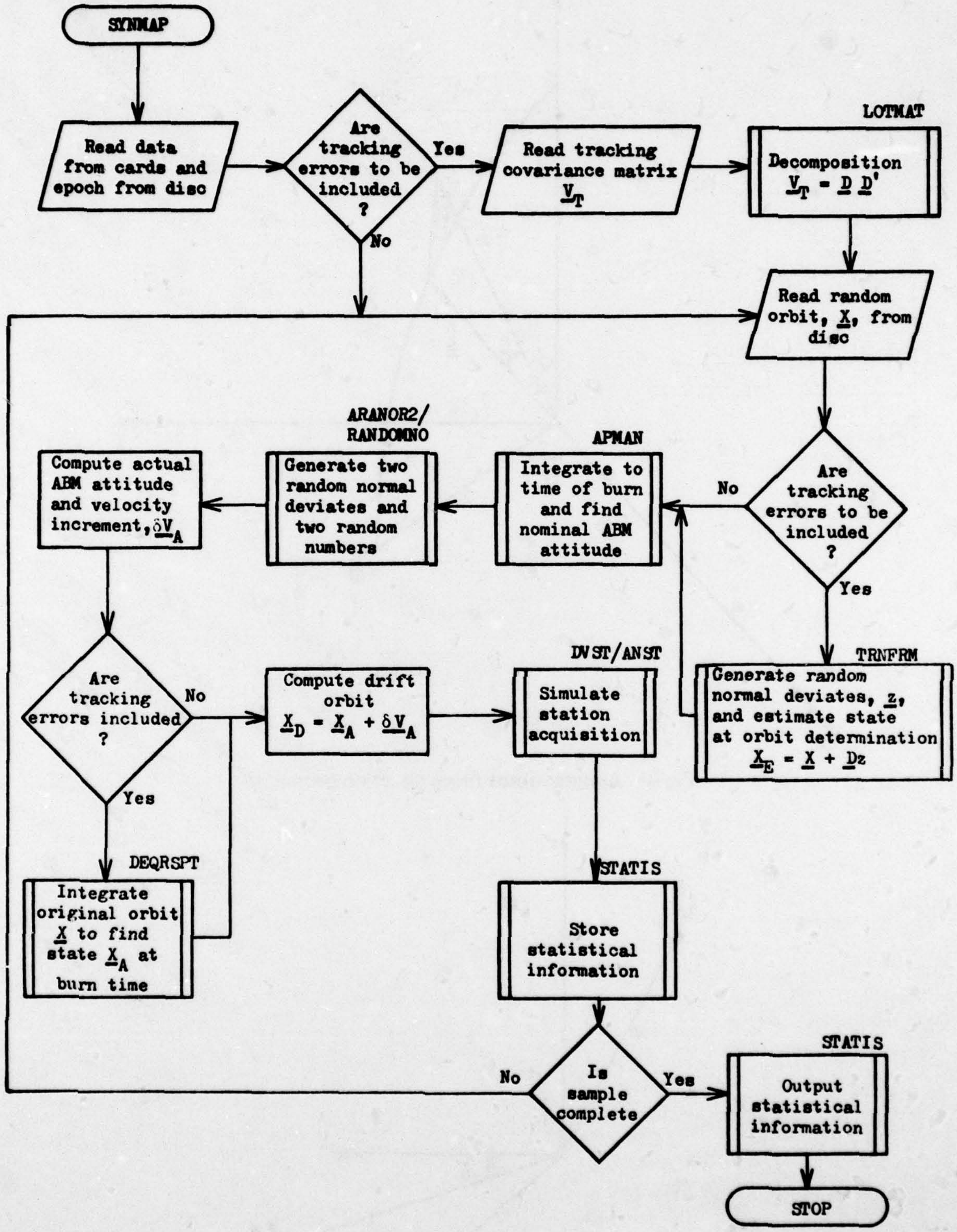


Fig 3 SYNMAP Flowchart

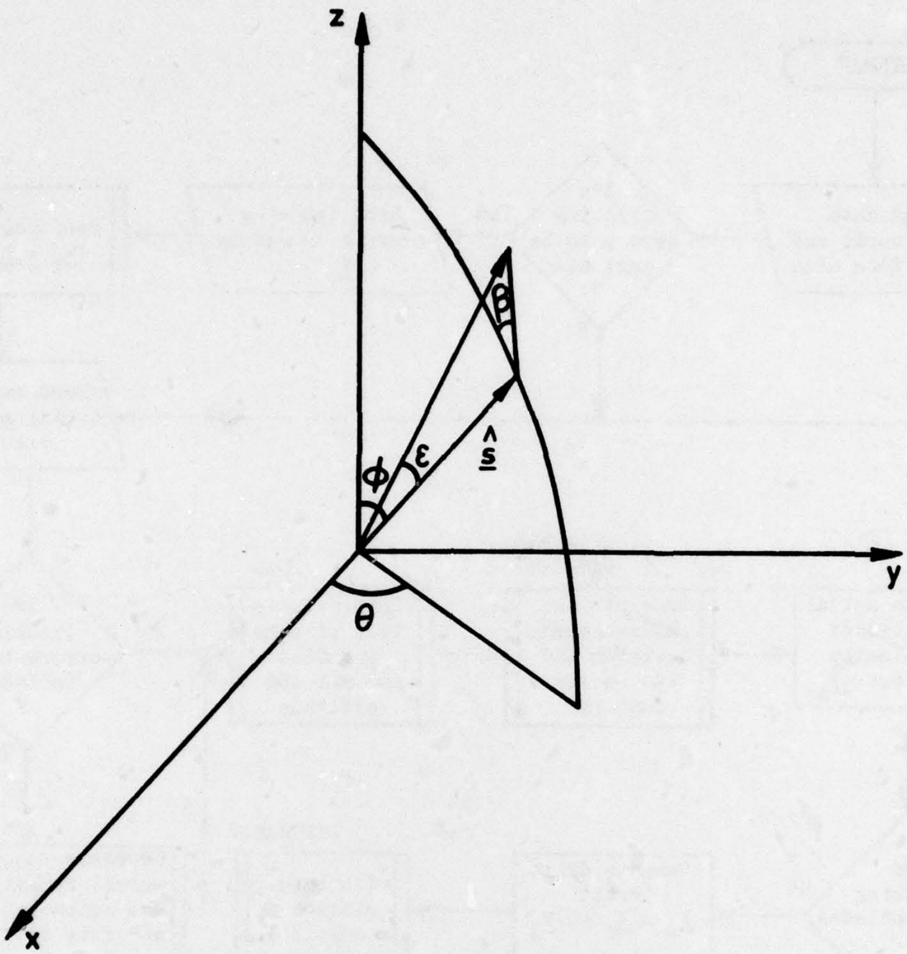


Fig 4 Apogee motor firing direction geometry

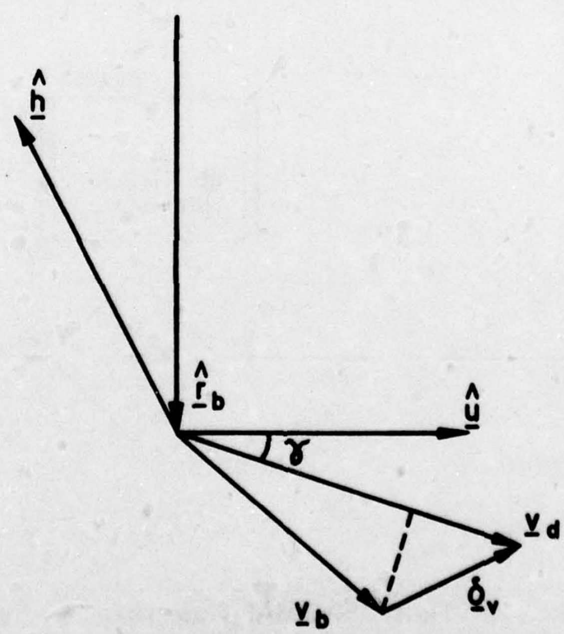


Fig 5 Apogee motor firing strategy for fixed flight path angle

```
43445 0.0 34.0 56.0 0.95
 1000 2 0 0
0.0 0.00461689 150.0 150.0
10.25532912 0.0 117.4089 6563.363 -2.3234 7.0
1 -0.0012648 -0.0015655 -0.0025252
 1.68459822E+02-2.36945325E-01 4.34254639E-01
-1.05337572E+04 1.82701771E-02-2.91304240E-02
-2.36945325E-01 1.13359136E-02-3.73616831E-06
 2.34053026E+01-1.32153668E-03 2.29848297E-03
 4.34254639E-01-3.73616831E-06 1.06082481E-02
-6.76526159E-01-6.08169840E-06-6.20077460E-05
-1.05337572E+04 2.34053026E+01-6.76526159E-01
 1.27337905E+07-5.26943999E+00 8.27593956E+00
 1.82701771E-02-1.32153668E-03-6.08169840E-06
-5.26943999E+00 2.40459017E-03-3.83190211E-03
-2.91304240E-02 2.29848297E-03-6.20077460E-05
 8.27593956E+00-3.83190211E-03 6.80415065E-03
```

Time card

Control card

Nominal orbit card

Gross attitude manoeuvre card

Covariance matrix

Fig 6 POINT2 data deck

```
5 3.0 0 0 1
QFPA 1.79587 2.7 209.8 0.0
DVSI 21.0 2.0 40.0 2.35 3.05
0.0 0 0.0 0.0 0.0
```

Control card

Attitude manoeuvre card

Station acquisition card

Perturbation card

Fig 7 SYNMAP data deck

```
                              SYNCHRONOUS MISSION ANALYSIS

DRAG IS NOT INCLUDED          LUNI-SOLAR PERTURBATIONS ARE NOT  INCLUDED     SOLAR RADIATION PRESSURE IS NOT INCLUDED

NOMINAL DELTAV = 1.79587KM/S      DESIRED INCLINATION = 2.70 DEG      DESIRED RIGHT ASCENSION = 209.8 DEG
TOTAL TIME FOR STATION ACQUISTION = 21.0 DAYS      TRACKING TIME = 2.0 DAYS      STATION LONGITUDE =  40.0 DEG

    1)        321.150    -19.743  41645.1581  0.01400       2.711   209.842    -86.505    94.793    367.762    359.511
EASTWARD DRIFT     6.777      0.000     19.241     19.241     21.533     0.000     21.533
WESTWARD DRIFT    17.524     68.993     49.753    118.746     21.533     0.000    118.746

    2)        321.775    -19.483  41985.0243  0.0062771     2.713   209.846    -66.725    75.883    363.306    355.794
EASTWARD DRIFT     2.320      0.000      6.588      6.588      9.650     0.000      9.650
WESTWARD DRIFT    16.865     56.672     47.884    102.356      9.650     0.000    102.356

    3)        320.640    -19.521  42007.2663  0.0146653     2.675   209.692    -34.173    43.070    363.017      2.225
EASTWARD DRIFT     2.032      0.000      5.769      5.769     22.545     0.000     22.545
WESTWARD DRIFT    17.179     54.544     48.775    103.318     22.545     0.000    103.318

    4)        321.900    -19.310  41389.0006  0.0181547     2.976   210.721   -102.818   109.877    371.182    359.813
EASTWARD DRIFT    10.196      0.000     28.949     28.949     27.910     0.000     28.949
WESTWARD DRIFT    17.882     79.719     50.770    130.490     27.910     0.000    130.490

    5)        321.393    -19.809  41506.5299  0.0137586     2.599   209.405   -108.138   117.027    369.606    358.275
EASTWARD DRIFT     8.621      0.000     24.476     24.476     21.151     0.000     24.476
WESTWARD DRIFT    17.641     74.563     50.087    124.650     21.151     0.000    124.650



                             MEAN                 STANDARD DEVIATION

   ABM ATTITUDE -
    RIGHT ASCENSION       321.372      DEGREES       0.453      DEGREES
    DECLINATION           -19.573      DEGREES       0.181      DEGREES

    SOLAR ASPECT ANGLE    100.890      DEGREES       0.428      DEGREES                CONSTRAINT VIOLATED      0 TIMES
   ELEVATION ANGLE        -11.140      DEGREES       0.445      DEGREES
   SEMI-MAJOR AXIS      41706.596      KM          250.037      KM
   ECCENTRICITY             0.0133725                0.0038842
   INCLINATION              2.735      DEGREES       0.127      DEGREES
   RIGHT ASCENSION        209.901      DEGREES       0.440      DEGREES
   INITIAL LONGITUDE      359.123      DEGREES       2.100      DEGREES
   TOTAL DELTAV            21.431      M/SEC         6.416      M/SEC


   DELTAV DISTRIBUTION -

         DV <   5 M/SEC        0
         DV <  10 M/SEC        1
         DV <  15 M/SEC        0
         DV <  20 M/SEC        0
         DV <  25 M/SEC        3
         DV <  30 M/SEC        1
         DV <  35 M/SEC        0
         DV <  40 M/SEC        0
         DV <  45 M/SEC        0
         DV <  50 M/SEC        0
         DV <  55 M/SEC        0
         DV <  60 M/SEC        0
         DV <  65 M/SEC        0
         DV <  70 M/SEC        0
         DV <  75 M/SEC        0
         DV <  80 M/SEC        0
         DV <  85 M/SEC        0
         DV <  90 M/SEC        0
         DV <  95 M/SEC        0
         DV < 100 M/SEC        0
         DV > 100 M/SEC        0
```

Fig 8 Sample lineprinter output from SYNMAP

REPORT DOCUMENTATION PAGE

Overall security classification of this page

UNCLASSIFIED

As far as possible this page should contain only unclassified information. If it is necessary to enter classified information, the box above must be marked to indicate the classification, e.g. Restricted, Confidential or Secret.

1. DRIC Reference (to be added by DRIC)	2. Originator's Reference	3. Agency Reference	4. Report Security Classification/Marking
	RAE TR 77179	N/A	UNCLASSIFIED

5. DRIC Code for Originator	6. Originator (Corporate Author) Name and Location
850100	Royal Aircraft Establishment, Farnborough, Hants, UK
5a. Sponsoring Agency's Code	**6a. Sponsoring Agency (Contract Authority) Name and Location**
N/A	N/A

7. Title
The synchronous mission analysis program SYNMAP.

7a. (For Translations) Title in Foreign Language

7b. (For Conference Papers) Title, Place and Date of Conference

8. Author 1. Surname, Initials	9a. Author 2	9b. Authors 3, 4	10. Date	Pages	Refs.
Palmer, M.D.	Cook, G.E.		December 1977	106	15

11. Contract Number	12. Period	13. Project	14. Other Reference Nos.
N/A	N/A		Space 540

15. Distribution statement
(a) Controlled by – [redacted]
(b) Special limitations (if any) –

16. Descriptors (Keywords) (Descriptors marked * are selected from TEST)
Communications satellites. Synchronous satellites. Mission analysis. Elliptic orbits. Transfer orbits. Drift orbits. Station acquisition. Apogee motor firing.

17. Abstract

A detailed description is given of the computer program SYNMAP, which uses a stochastic simulation method to determine the total velocity increment required for the station acquisition phase of a synchronous satellite orbit mission. The program takes account of errors due to launch vehicle injection, satellite tracking and apogee motor burn. A description is also given of the program POINT2, which may be used to generate the set of random transfer orbits required by SYNMAP.